PROBABILISTIC MODELING OF NEURAL DATA FOR ANALYSIS AND SYNTHESIS OF SPEECH

A Thesis
Presented to
The Academic Faculty

by

Brett Alexander Matthews

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
December 2012

PROBABILISTIC MODELING OF NEURAL DATA FOR ANALYSIS AND SYNTHESIS OF SPEECH

Approved by:

Dr. Mark A. Clements, Adviser
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. David V. Anderson
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Elliot Moore II
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Jonathan Brumberg
Department of Speech-Language-Hearing
University of Kansas

Dr. William B. Hunt
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: August 17, 2012

ACKNOWLEDGEMENTS

First and foremost, and with the utmost sincerity, I thank our Beneficent and Omnipotent God, with whose help all is achievable.

My father Roosevelt Matthews, my mother Brenda Matthews and my brother Scott Matthews: I cannot thank you enough for your many prayers, support, encouragement, and love. I could not do it without you.

I thank especially my research adviser Prof. Mark Clements for valuable support through the process while still giving me enormous freedom to shape the direction of the work. To my first research adviser, the late Prof. D. Scott Wills, I thank you much for your guidance, mentorship, kind words and support; you were one of a kind and you are truly missed.

On the Georgia Tech faculty, I am especially grateful to Prof. Linda Wills for her support and advice, and to Prof. Chin-Hui Lee for many valuable and insightful words of advice regarding research and other matters. I especially thank Dean Gary S. May for providing me and many others with valuable inroads into graduate study and Prof. Mark J.T. Smith for being a great example. Many thanks to Prof. Biing-Hwang (Fred) Juang, Prof. Jim McClellan and the rest of the ECE2025 faculty; it was a pleasure to serve as head TA. I would like to thank my committee Prof. David Anderson, Prof. Elliot Moore II, Prof. William Hunt, and especially my frequent collaborator Prof. Jonathan Brumberg of the University of Kansas, for their valuable insights, comments and questions. Among the ECE staff, I am grateful for the efforts of Jill Auerbach, Marilou Mycko, Tasha Torrence and Chris Malbrue.

Thanks to my friends Dr. Gerald DeJean, Gerald Coleman, Gavin Bell, Travis

Smith, M. Janae Lane, Esq., Isake McLeod, Dr. Otis Smart, Dr. Cynthia Vance-Harris, Dr. Jacqueline Fairley, Dr. Senyo Apewokin, Fred Green, Dr. Ashley Johnson-Long, Dr. Sekou Remy, Dr. Karolyn Babalola, Dr. Roshawnna Novellus, Dr. Clyde Lettsome, Dr. Tammy McCoy, Jonathan Kim, Daimia T. Gunning, Glenda M. McKnight, and Latasha McAlpine, Esq. Thanks to my labmates, Dr. Yeongseon Kim, Dr. Ryan Palkki, Dr. Kaustubh Kalgaonkar, Aditya Joshi, Dr. Jongmyon Kim, Dr. Yu Tsao, Dr. Jeremy Reed, Dr. Marco Siniscalchi, Dr. Qiang Fu, Dr. Jinyu Li, Dr. Antonio Moreno-Daniel, Dr. Sibel Yaman, Hrishikesh Rao and Teresa Sanders. Thanks to the CSIP staff Patricia Dixon, Cordai Ferrar, Stacie Speights and Jennifer Lunsford.

Thanks especially to my mentor at IBM Dr. Bhuvana Ramabhadran for her great advice and support and for being a great leader. Thanks to my collaborators, mentors and friends at IBM and MIT Lincoln Labs: Dr. Raimo Bakis, Dr. Ellen Eide, Dr. Tara Sainath, Dr. John Pitrelli, Dr. Michael Picheny, Dr. John-David Wellman, Dr. Nick Malyska, and Dr. Thomas Quatieri.

I thank Dr. Phillip Kennedy of Neural Signals Inc. and Prof. Frank Guenther of Boston university for the opportunity to work on such an exciting project. Finally, I would like to express to Erik Ramsay and his father Eddie my thanks for their great patience, and my highest admiration and respect.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This research consists of probabilistic modeling of speech audio signals and deep-brain neurological signals in brain-computer interfaces. A significant portion of this research consists of a collaborative effort with Neural Signals Inc., Duluth, GA, and Boston University to develop an intracortical neural prosthetic system for speech restoration in a human subject living with Locked-In Syndrome, i.e., he is paralyzed and unable to speak. The work is carried out in three major phases. We first use kernel-based classifiers to detect evidence of articulation gestures and phonological attributes speech audio signals. We demonstrate that articulatory information can be used to decode speech content in speech audio signals. In the second phase of the research, we use neurological signals collected from a human subject with Locked-In Syndrome to predict intended speech content. The neural data were collected with a microwire electrode surgically implanted in speech motor cortex of the subject's brain, with the implant location chosen to capture extracellular electric potentials related to speech motor activity. The data include extracellular traces, and firing occurrence times for neural clusters in the vicinity of the electrode identified by an expert. We compute continuous firing rate estimates for the ensemble of neural clusters using several rate estimation methods and apply statistical classifiers to the rate estimates to predict intended speech content. We use Gaussian mixture models to classify short frames of data into 5 vowel classes and to discriminate intended speech activity in the data from non-speech. We then perform a series of data collection experiments with the subject designed to test explicitly for several speech articulation gestures, and decode the data offline. Finally, in the third phase of the research we develop

an original probabilistic method for the task of spike-sorting in intracortical brain-computer interfaces, i.e., identifying and distinguishing action potential waveforms in extracellular traces. Our method uses both action potential waveforms and their occurrence times to cluster the data. We apply the method to semi-artificial data and partially labeled real data. We then classify neural spike waveforms, modeled with single multivariate Gaussians, using the method of minimum classification error for parameter estimation. Finally, we apply our joint waveforms and occurrence times spike-sorting method to neurological data in the context of a neural prosthesis for speech.

CHAPTER I

INTRODUCTION

Interpersonal communication is essential to the human experience. Among the many modes and methods of communication humans have developed, speech is arguably the most natural and is routinely preferred by parties in close proximity. The activity of spoken communication is an integral part of all human cultures and is executed at a level of sophistication and complexity that is uniquely human.

The mechanisms of speech production at the physiological, psychological and neurological levels are each quite complex in their own right, more so than most speakers are probably aware. Speech production involves the coordinated activity of the natural speech apparatus in the mouth, nose, throat and lungs, as well as several distinct regions of the brain controlling language, word formation and motor activity. The analysis of speech production and communication is therefore of great importance in many fields of study, including Engineering, Neuroscience, Biomedical Science, Psychology, and Linguistics.

When the ability to speak spontaneously becomes severely impaired for a person, the quality of his or her life is greatly compromised. Much like speech production itself, the nature of impairments to speech production is also largely physiological, psychological or neurological, and therapies and treatments vary along these lines as well. Given the considerable complexity of the speech production process, developing solutions or treatments for severe speech impairments remains a significant challenge. The research presented in this work consists of digital signal processing and statistical modeling methods applied to audio signals and speech-related neurological signals. A significant portion of the work is concerned with addressing certain types of severe

neurological speech impairments through the use of a brain-computer interface.

Under normal conditions, the central nervous system coordinates the motion of the natural speech apparatus, i.e., a network of muscles and organs (most of which have other functions such as eating and breathing) in the oral, nasal, thoracic and abdominal cavities. Traumatic brain injuries and certain neurological disorders, such as Amyotrophic Lateral Sclerosis (ALS), are examples of severe impairments at the neurological level of speech production which can cause a person to become both unable to speak and fully paralyzed. This especially difficult set of conditions is collectively known as "Locked-In Syndrome." [69]. In certain cases of Locked-In Syndrome, even though control of the speech apparatus is effectively severed, some cortical activity in regions of the brain related to speech may remain intact. In these cases, persons living with Locked-In Syndrome may benefit from a *neural prosthesis for speech restoration* based on an intracortical brain-computer interface (BCI).

Neural prosthetic systems with intracortical brain-computer interfaces (BCIs) have received much attention in recent years. These systems are designed to allow humans and non-human primates to control computers, robotic arms and other devices using neurological signals extracted from a population of cortical neurons. In addition to enhancing our understanding of the brain, BCIs offer the possibility of restoring mobility and basic communication needs to persons living with severe motor and speech impairments.

Research in intracortical neural prosthetic systems can roughly be grouped into two broad categories: motor control systems and communication systems [53]. Recent work in both of these areas has produced some striking results with non-human primates and human volunteers. Some notable motor control studies with primates include [78] where a monkey was taught to control the position of a cursor on a computer screen and [86] where monkeys were trained to control a multi-jointed robotic arm for self-feeding. Neural prostheses for *communication* are particularly relevant

and important to humans living with the effects of severe neurological disorders and traumatic brain injuries. In several important studies, human subjects living with Locked-In Syndrome were able to operate communication devices using only cortical neural signals. In [43, 44], patients learned to operate a virtual keyboard and a computer-simulated on-screen "hand" based on local field potentials of cortical neural signals, while subjects in another study [33] learned to control a computer cursor via a visual feedback mechanism.

A significant portion of the research we present here consists of our contributions to a real-time, intracortical neural prosthetic system for speech communication with a human subject. The study, the very first of its kind [10, 28] involved a 26-year-old male subject living with Locked-In Syndrome due to a brain stem stroke following neurological trauma. Cortical neurological signals were collected from the subject, whom we refer to as "ER," through a microwire electrode surgically implanted in his brain in **speech motor cortex**. Our work in this study consists of collecting neurological signals extracted during imagined speech production, and classifying and decoding these data as well as previously collected data by our colleagues in the same study.

Our research consists of probabilistic models of speech audio signals and cortical neurological signals for the purpose of a neural prosthesis for speech production. The research is carried out in three distinct thrusts:

1. The first research thrust consists of background work in which we investigate methods of detecting phonological attributes, including speech motor or articulatory gestures, in speech audio signals, along with lessons learned.

2. The second major thrust consists of our work in collecting and decoding deep-brain neurological signals related to speech motor and articulation gestures from a human patient living with Locked-In Syndrome. We use probabilistic methods

of classification and decoding built on the rich foundation of research in statistical pattern recognition and signal processing for the well-known automatic speech recognition (ASR) task.

3. Finally, in the third research thrust, we develop probabilistic methods for the task of automatic action potential identification and classification (often called "spike sorting") applicable to brain-computer interface (BCI) systems in general and, specifically to a neural prosthesis for speech. We develop a novel probabilistic framework for jointly modeling the firing times and waveform shapes of observed neural action potentials in neurological signals. We then apply methods of discriminative training to action potential waveform models using the minimum classification error (MCE) criterion for spike classification in brain computer interfaces.

1.1 Objective of the Research and Design Hypotheses

The objective of the research is to develop probabilistic models of cortical neural data for stimulus decoding and action potential identification in the context of a neural prosthesis for speech. The work consists of decoding actual and intended speech motor production in speech audio signals and speech-related neurological signals, respectively, and automatic action potential identification in recorded traces of extracellular potentials in brain-computer interfaces.

The research is carried out in three distinct thrusts, as identified in the previous section. In the remainder of this section, for each major thrust of the research, we identify the principle hypothesis and its corresponding design assumptions.

1. Evidence of articulation in speech audio signals can be used to decode intended speech content.

Our approach to the neural speech prosthesis problem is to infer intended speech

content from cortical neurological signals related to speech motor activity and articulation gestures. As a result, we do not expect to find a direct representation of intended speech content in these signals. Rather, we expect to find encoded evidence of activations to the set of speech articulators such as the teeth, tongue, jaw, lips and glottis. For this approach to succeed, however, it must be examined whether evidence of speech motor or articulatory gestures themselves comprise a sufficient information source for decoding speech and, if so, to what extent.

In the first research thrust, we motivate the speech motor approach to a neural speech prosthesis by detecting articulatory and other phonological information in speech *audio* signals for automatic speech recognition. Speech production can be categorized according to the articulators used and the manner and place in which they were used [13]. The /s/ sound, for example, is a voiceless frication produced by creating a partial constriction of air flow using the tip of the tongue at the alveolar ridge. *Voiced*, *fricative*, *lingual*, and *alveolar* can be considered "attributes" of speech production corresponding to the /s/ sound. Our approach is to train statistical classifiers to detect phonological attributes of the speech signal and to recognize speech based on the decoded result. With this approach, we aim to show that statistical modeling of articulatory information can be used to decode intended speech content, with comparable accuracy to state-of-the art automatic speech recognition systems.

2. Imagined attempts at speech activity can be decoded from neurological signals in speech motor cortex.

The second thrust of the research, i.e., classifying and decoding deep-brain neurological signals into intended speech sounds, depends on several important assumptions. All of our experiments in this phase of the research are carried out using an existing physical infrastructure which includes an electrode surgically implanted in speech motor cortex in the brain of a human subject living with Locked-In Syndrome.

Using a passive electrode, we assume that action potential events are reliably detected in the electrical activity of a population of neurons in the presence of various possible sources of incumbent noise. While microwire electrode traces easily capture the firing activity of single and multiple neurons, electromagnetic interference, the activity of far field neurons, and many other sources of system noise contribute to variation and errors in neural spike acquisition. In any intracortical BCI, care should be taken to eliminate noise and mitigate its effects.

We must also assume that the firing activity of neurons in the vicinity of the electrode encodes intended speech or speech motor activity in some discernible way, and that the subject has control over this. The organization of speech function in the brain is very complex, with separate brain regions controlling language, phonetic word formation, comprehension, and speech motor control of the articulators. All of these interdependent operations, as well as some form of external feedback, are involved in normal speech function. In the design of the physical infrastructure for this study [10, 3], considerable care was taken to determine an implant location related to speech *motor* function. This was chosen so that, among other reasons, the approach to the speech prosthesis problem could be modeled after many successful motor control studies with non-human primates [78, 86] as well as humans [33]. In any case, we must assume either that the subject retains conscious control of speech function or that he can regain it through experiment and practice with the apparatus.

3. Temporal modeling of neural firing times can be used to improve action potential classification performance in BCIs.

Finally, classifying individual neurons based on action potential spikes in an extracellular trace is a critical component of the front-end signal processing stage of a BCI system. All subsequent processing depends on the result of this operation, often called "spike-sorting." In the final thrust of the research, we present a novel

probabilistic method for spike-sorting using both action potential spikes and their corresponding firing times. For this method to succeed, however, we must assume that the firing activity of individual neurons *potentially* contains information related to both the stimulus or activity of interest and for discriminating between neurons.

1.2 Organization

In Chapter 3, we discuss contributions we made toward a new paradigm for ASR based, in part, on detecting distinctive attributes of speech audio signals. Many of these speech attributes are articulatory and phonemic in nature, and related to speech motor production. Our specific contributions to a real-time neural prosthesis for speech restoration are described in Chapters 4 and 5. In Chapter 4 we formally define a neural speech prosthesis and advocate decoding methods based on the well-known ASR problem. In Chapter 5 we describe a series of real-time experiments in collecting neural data from a human subject who suffered a traumatic brain injury and lives with Locked-In Syndrome. We discuss a real-time software framework for data-collection, and the results of applying our decoding method to the collected data.

Finally, we present a novel, automatic approach to the task known as "spike-sorting," which is an important part of the *Front-End Signal Processing* component of an intracortical neural speech prosthesis. Given a trace of the superimposed electrical activity of a population of neurons in the vicinity of an electrode, spike-sorting is the task of detecting, identifying and classifying each neuron in the population based on its characteristic, observed electrical activity. In Chapter 6, we describe a novel method for incorporating neuronal firing times into the spike-sorting task.

CHAPTER II

BACKGROUND

2.1 Statistical Modeling for Automatic Speech Recognition

Research in systems for automatic speech recognition (ASR), understanding, and various related tasks (e.g., machine translation [9] and audio search [15, 90, 57]) has a long history, and the performance of these systems has improved significantly over the course of many years. Generally, these systems are tasked to take input speech signals and decode them into a sequence of words or other speech- or language-related symbols. The most successful approaches to ASR are strongly rooted in the theory and practical application of statistical modeling. In fact, many innovations first presented in the context of ASR research have been widely adopted in other areas of statistical modeling as well [1] .

Generally, given a random variable $\mathcal{X}$, realized as an $N \times D$ matrix $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ of N observations, each of dimension D, we use statistical modeling approaches to both describe the data in $\mathbf{X}$ and to make important inferences about $\mathbf{X}$ with respect to other variables, and vice-versa. For the ASR task, $\mathbf{X}$ is typically a collection of multivariate, continuous-valued, frequency-based parameterizations of the input speech signal. The discrete-valued random variable $\mathcal{W}$ typically represents the set of possible sequences of words or other symbols, of which $\mathbf{w}$ is a realization. The goal of typical statistical approaches to the decoding task in ASR is to find the *maximum a posteriori* sequence $\hat{\mathbf{w}}$ given the observed speech data in $\mathbf{X}$, i.e.,

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} P(\mathbf{w}|\mathbf{X}) = \arg\max_{\mathbf{w}} \frac{P(\mathbf{X}|\mathbf{w})P(\mathbf{w})}{P(\mathbf{X})}, \tag{1}$$

where we use Bayes' rule to decompose the posterior probability $P(\mathbf{w}|\mathbf{X})$ into the

predictive likelihood $P(\mathbf{X}|\mathbf{w})$, the prior probability $P(\mathbf{w})$ and the dataset likelihood $P(\mathbf{X})$; this form is typically much more convenient. Since $P(\mathbf{X})$ does not vary with $\mathbf{w}$, we can write

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} P(\mathbf{X}|\mathbf{w})P(\mathbf{w}). \tag{2}$$

In the ASR problem, the predictive distribution $P(\mathbf{X}|\mathbf{w})$ is often called the "acoustic model"; it is a generative model of the sequence of multivariate, parameterized acoustic observations in $\mathbf{X}$. The prior probability $P(\mathbf{w})$ gives the probability of the sequence of words or symbols in $\mathbf{w}$ and is often called the "language model."

The acoustic model $P(\mathbf{X}|\mathbf{w})$ should be effective for both describing and decoding observed speech data. To assume that the acoustic observations in $\mathbf{X}$ are independent and identically distributed (i.i.d.), as is common in many statistical modeling problems, is inadequate for ASR and related tasks. It is well-known that the statistics of acoustic speech signals, as well as all commonly used parameterizations of speech signals, are non-stationary with respect to time and, perhaps more significantly, vary with respect to the set of speech sounds [22]. For these reasons, hidden Markov models (HMMs), which provide a framework for non-stationary, generative probabilistic modeling, have become the dominant approach to statistical acoustic modeling for ASR.

2.2 Hidden Markov Models

In this section, we provide a comprehensive review of hidden Markov models (HMMs) and Gaussian mixture models (GMMs). Our probabilistic approach to modeling and decoding of neurological signals makes extensive use of GMMs and HMMs, building strongly on the framework of statistical modeling for automatic speech recognition. We use GMMs to make unsupervised clusterings of neural action potential waveforms (cf. Section 2.3 and Chapter 6) and as a statistical classifier for neural firing rate

estimates (Section 2.3.3 and Chapters 4, and 6). In Chapter 5, we use a hidden Markov model to decode neurological signals into intended speech.

In the remainder of this section, we discuss HMM and GMM likelihood models, including methods for parameter estimation and inference. The models are discussed in the context of automatic speech recognition but with sufficient generality to be applied more widely.

Given an input speech signal $s(t)$, let $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ be a sequence of N multivariate observations, each of dimension D; $\mathbf{X}$, as we define it, is the result of applying some transformation to the signal $s(t)$. The operation applied to $s(t)$ to produce the set of feature observations in $\mathbf{X}$ is usually a frequency-based transformation on short, possibly overlapping segments of the signal in $s(t)$, equally spaced in time. Some of the more widely used feature transformations in ASR are the Mel-frequency Cepstral Coefficients (MFCC) features [18], and the Perceptual Linear Prediction (PLP) features [32]. The decoding task for ASR is to infer a time-aligned sequence of discrete states $\mathbf{w} = \{w_1, \cdots, w_N\}$ given $\mathbf{X}$. Typically, multiple HMM states are assigned to each phone, word, or subword unit.

Hidden Markov models are parametric, non-stationary, latent variable likelihood models [5]. HMMs are especially appropriate for modeling and decoding non-stationary, labeled data such as speech signals. To motivate the discussion of HMMs as a likelihood model we can express $P(\mathbf{X}; \lambda)$, the data-set likelihood for $\mathbf{X}$ as follows:

$$P(\mathbf{X}\,;\lambda) = \sum_{\mathbf{w}} P(\mathbf{X}, \mathbf{w}\,;\lambda) \tag{3}$$

$$= \sum_{\mathbf{w}} P(\mathbf{X}|\mathbf{w}\,;\lambda) P(\mathbf{w}\,;\lambda) \tag{4}$$

where λ is the full set of model parameters, and $\mathbf{w}$ is a sequence of states corresponding with observations in $\mathbf{X}$.

To express the likelihood in (3) and (4) using an *HMM*, we introduce the HMM

parameter set $\lambda = \{\boldsymbol{\pi}, \mathbf{A}, \{\theta_j\}_{j=1}^{L}\}$, where L is the total number of possible states in the HMM, A is an $L \times L$ matrix, and π is a vector of dimension L. The parameters $\boldsymbol{\pi}$ and $\mathbf{A} = \{a_{ij}\}$ govern transitions between states such that the likelihood $P(\mathbf{w}\ ;\lambda)$ (note that this is the second term in (4)) for a sequence of states $\mathbf{w}$ is given by

$$P(\mathbf{w};\lambda) = P(\mathbf{w}\ ;\mathbf{A},\boldsymbol{\pi}) = \pi_{w_0} a_{w_0 w_1} a_{w_1 w_2} \cdots a_{w_{N-1} w_N} \tag{5}$$

where a_{ij} is the probability of transitioning from state i to state j, w_t is the state of the HMM at time t, and $\boldsymbol{\pi}$ is the initial state distribution at time $t = 0$, i.e., $\pi_i = P(w_0 = i)$.

The first term in (4), the complete data-set likelihood $P(\mathbf{X}|\mathbf{w};\lambda)$, given a hypothesized state sequence $\mathbf{w}$, depends only the parameters $\{\theta_j\}_{j=1}^{L}$. The likelihood is given by

$$P(\mathbf{X}|\mathbf{w}\ ;\lambda) = P(\mathbf{X}|\mathbf{w}\ ;\{\theta_j\}_{j=1}^{L}) = \prod_{t=1}^{N} p(\mathbf{x}_t\ ;\theta_{w_t}). \tag{6}$$

Though the HMM as a whole is a non-stationary model, observed data corresponding to any given single state i of an HMM are typically expressed with a stationary probability distribution as in the RHS of (6). For continuous-valued data, it is common to use a mixture of Gaussians with parameters $\theta_i = \{c_{i,j=1}^{K}, \boldsymbol{\mu}_{i,j=1}^{K}, \boldsymbol{\Sigma}_{i,j=1}^{K}\}$ such that $p(\mathbf{x}\ ;\theta_i) = \sum_k c_k \mathcal{N}(\mathbf{x};\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)$. The complete data-set likelihood $P(\mathbf{X};\lambda)$ obtained by substituting the expressions for $P(\mathbf{X}|\mathbf{w}\ ;\lambda)$ and $P(\mathbf{w}\ ;\lambda)$ into (4), is then

$$P(\mathbf{X}\ ;\lambda) = \sum_{\mathbf{w}} \pi_{w_0} \prod_{t=1}^{N} a_{w_{t-1} w_t} p(\mathbf{x}_t\ ;\theta_{w_t}). \tag{7}$$

Note that evaluating the likelihood in (7) directly requires a sum over L^N possible sequences, which is computationally intractable on large data-sets. The computation can be done in $L^2 \cdot N$ steps using an efficient procedure called the Forward Algorithm [73] .

2.2.1 Gaussian Mixture Models and Expectation Maximization

As discussed in Section 2.2, it is common to use a mixture of Gaussians for the likelihood model for each state in an HMM. Generally, the likelihood $p(\mathbf{x};\theta)$ for a "mixture model" of continuous-valued, multivariate data in $\mathbf{x}$ is as expressed below

$$p(\mathbf{x};\theta) = \sum_{i=1}^{K} c_i f(\mathbf{x};\mathbf{b}_i). \tag{8}$$

The likelihood $p(\mathbf{x};\boldsymbol{\theta})$ for a mixture model is the weighted sum of the likelihoods of K mixture components $f(\mathbf{x};\mathbf{b}_i)_{i=1}^{K}$ with mixture weights $c_i{}_{i=1}^{K}$ The weights c_i must sum to 1, and each mixture component $f(\mathbf{x};\mathbf{b}_i)$ must be a valid parametric probability density or mass function; with these two conditions, it can be shown that the mixture model $p(\mathbf{x};\boldsymbol{\theta})$ is a valid probability density or mass function. Also, the mixture weight c_i can be interpreted as the a priori probability of mixture component i. For Gaussian mixture models, the form of each mixture component $f(\mathbf{x};\mathbf{b}_i)$ is a multivariate normal as given below

$$\begin{aligned} f(\mathbf{x};\mathbf{b}_i) =& \mathcal{N}(\mathbf{x};\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i) && (9)\\ =& \frac{1}{(2\pi)^{D/2}\left|\boldsymbol{\Sigma}_i\right|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i(\mathbf{x}-\boldsymbol{\mu})\right\}, && (10) \end{aligned}$$

where $\mathbf{x}$ is of dimension D.

The parameter set for a Gaussian mixture model with K mixture components is $\theta = \{c_{i=1}^{K}, \boldsymbol{\mu}_{i=1}^{K}, \boldsymbol{\Sigma}_{i=1}^{K}\}$. Unlike a single Gaussian density, no closed-form expression exists to find the maximum likelihood parameters for the mixture of Gaussians density. It is possible, however, to formulate an iterative parameter estimation procedure, such that the log-likelihood $l(\theta)$ is guaranteed to increase on each iteration of the procedure, i.e., $l(\theta_{t+1}) > l(\theta)$.

Let $l(\theta)$ be the log-likelihood of the mixture model with dataset $\mathbf{X} = \mathbf{x}_{i=1}^{N}$ and parameter set θ

$$l(\theta) = \log\left\{\prod_{i=1}^{N} p(\mathbf{x}_i;\theta)\right\} = \log\left\{\prod_{i=1}^{N}\sum_{j=1}^{K} p(\mathbf{x}_i, m_j = j;\theta)\right\} \tag{11}$$

$$= \sum_{i=1}^{N} \log \sum_{j=1}^{K} p(\mathbf{x}_i, m_j = j;\theta), \tag{12}$$

where the discrete-valued variable m_i ($1 \leq m_i \leq K$) indicates one of K mixture components. Continuing,

$$l(\theta) = \sum_{i=1}^{N} \log \sum_{j=1}^{K} p(m_i = j|\mathbf{x}_i;\theta_t)\frac{p(\mathbf{x}_i, m_i = j;\theta)}{p(m_i = j|\mathbf{x}_i;\theta_t)} \tag{13}$$

Using Jensen's Inequality that $E[f(\mathbf{x})] \leq f(E[\mathbf{x}])$, where $E[\cdot]$ denotes an expectation operation and $f(\cdot)$ is any convex function, we can conclude the following since $\log(\cdot)$ is a convex function and the expression in (13) contains the log of an expectation

$$l(\theta) \geq \sum_{i=1}^{N}\sum_{j=1}^{K} p(m_i = j|\mathbf{x}_i;\theta_t)\log\frac{p(\mathbf{x}_i, m_i = j;\theta)}{p(m_i = j|\mathbf{x}_i;\theta_t)} \tag{14}$$

$$\geq \sum_{i=1}^{N}\sum_{j=1}^{K} p(m_i = j|\mathbf{x}_i;\theta_t)\log p(\mathbf{x}_i, m_i = j;\theta)$$

$$- \sum_{i=1}^{N}\sum_{j=1}^{K} p(m_i = j|\mathbf{x}_i;\theta_t)\log p(m_i = j|\mathbf{x}_i;\theta_t). \tag{15}$$

The second term in (15) does not vary with θ and does not need to be included in the parameter estimation procedure. Therefore, we define the auxiliary function $Q(\theta;\theta_t)$ (often called the "Q function") as follows

$$Q(\theta;\theta_t) = \sum_{i=1}^{N}\sum_{j=1}^{K} p(m_i = j|\mathbf{x}_i;\theta_t)\log p(\mathbf{x}_i, m_i = j;\theta). \tag{16}$$

If $Q(\theta;\theta_t)$ is concave with respect to θ, then $\arg\max_{\theta} Q(\theta;\theta_t)$ can be found by setting $\frac{\partial}{\partial\theta}Q(\theta;\theta_t)$ to zero. If we define θ_{t+1} as follows

$$\theta_{t+1} = \arg\max_{\theta} Q(\theta;\theta_t) \tag{17}$$

then $l(\theta_{t+1})$ is guaranteed to be greater than $l(\theta_t)$

2.2.2 Decoding for HMMs

The inference problem, i.e., to produce a decision about observed data based on an existing model, is often called "decoding" for time-varying data and models. For observed data in $\mathbf{X}$, HMM decoding is to find the state sequence $\hat{\mathbf{w}}$ to maximize the posterior probability $P(\mathbf{w}|\mathbf{X}) = P(\mathbf{X}, \mathbf{w})/P(\mathbf{X})$; since $P(\mathbf{X})$ does not vary with $\mathbf{w}$, we say

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} \; P(\mathbf{X}, \mathbf{w}; \lambda) \tag{18}$$

$$= \arg\max_{\mathbf{w}} \; \pi_{w_0} \prod_{t=1}^{N} p(\mathbf{x}_t \; ; \theta_{w_t}) \cdot a_{w_{t-1} w_t}, \tag{19}$$

where we have used expressions in (5) and (6). A direct search for the best of L^N possible sequences is computationally intractable. Instead, $\hat{\mathbf{w}}$ is found using the Viterbi Algorithm [73].

Let $\delta_j(t+1)$ be defined as the likelihood of the best sequence $\{w_i\}_{i=1}^{t+1}$ ending in state j up to, and including, time $t+1$, i.e., $\delta_j(t+1) = P(\{\mathbf{x}_i\}_{i=1}^{t+1}, \{w_i\}_{i=1}^{t}, w_{t+1} = j; \lambda)$. Exploiting that temporal modeling in the transition matrix $\mathbf{A}$ at time $t+1$ depends only on time t, we can formulate a recursive definition for $\delta_j(t+1)$ as follows

$$\delta_{t+1}(j) = \max_{1 \le i \le L} \left[\delta_t(i) a_{ij}\right] p(\mathbf{x}_{t+1} \; ; \theta_j). \tag{20}$$

The Viterbi algorithm uses $\delta_t(j)$ to keep track of the likelihood of the best state sequence up to time t, and a second variable $\psi_t(j)$ to keep track of the sequence itself. Upon termination at time $t = N$, the best sequence $\hat{\mathbf{w}}$ is found with a simple backtracking procedure using $\psi_t(j)$. The Viterbi algorithm is outlined below, including the definition of $\psi_t(j)$ and the backtracking procedure.

1. Initialization

$$\delta_1(i) = \boldsymbol{\pi}_i \cdot p(\mathbf{x}_1; \theta_i), \qquad 1 \leq i \leq L \tag{21}$$

$$\psi_1(i) = 0, \qquad 1 \leq i \leq L \tag{22}$$

2. Recursion

$$\delta_{t+1}(j) = \max_{1 \leq i \leq L} [\delta_t(i) a_{ij}]\, p(\mathbf{x}_{t+1}\ ; \theta_j), \qquad 1 \leq t \leq N-1,\ 1 \leq j \leq L \tag{23}$$

$$\psi_{t+1}(j) = \arg\max_{1 \leq i \leq L} [\delta_t(i) a_{ij}]\,, \qquad 1 \leq t \leq N-1,\ 1 \leq j \leq L \tag{24}$$

3. Termination

$$P(\mathbf{X}; \lambda) = \max_{1 \leq i \leq L} \delta_N(i) \tag{25}$$

$$\hat{w}_N = \arg\max_{1 \leq i \leq L} \delta_N(i) \tag{26}$$

4. Backtracking

$$\hat{w}_t = \psi_{t+1}(\hat{w}_{t+1}),\ t = N-1, \ldots, t = 2, t = 1 \tag{27}$$

$$\hat{\mathbf{w}} = (\hat{w}_1, \ldots \hat{w}_N) \tag{28}$$

2.2.3 Parameter Estimation for HMMs

Though the hidden Markov model framework is significantly more complex than a mixture of Gaussians, it is essentially a latent variable model for which an expectation-maximization procedure can be derived. In this section we briefly review the parameter estimation procedure for continuous-density hidden Markov models with mixtures of Gaussians for state emission probabilities. This is commonly called the Baum-Welch procedure [4].

The parameter set for the HMM is $\lambda = \{\boldsymbol{\pi}, \mathbf{A}, \{\theta_j\}_{j=1}^L\}$ where each $\theta_i = \{c_{i,j=1}^K, \boldsymbol{\mu}_{i,j=1}^K, \boldsymbol{\Sigma}_{i,j=1}^K\}$ as described in Section 2.2. The auxiliary or "Q function" for the whole parameter set λ is given by

$$Q(\lambda; \hat{\lambda}) = \sum_{all\ \mathbf{w}} \frac{P(\mathbf{X}, \mathbf{w}; \hat{\lambda})}{P(\mathbf{X}; \hat{\lambda})} \log P(\mathbf{X}; \hat{\lambda}) \tag{29}$$

We can substitute the expression for $P(\mathbf{X};\lambda)$ in (7) into (29) to obtain

$$Q(\lambda;\hat{\lambda}) = \sum_{all\ \mathbf{w}} \frac{P(\mathbf{X},\mathbf{w};\hat{\lambda})}{P(\mathbf{X};\hat{\lambda})} \left[\log \boldsymbol{\pi}_{w_0} + \sum_{t=0}^{N} \log a_{w_{t-1}w_t} + \sum_{t=1}^{N} \log p(\mathbf{x}_t;\theta_{\mathbf{w}_t})\right] \tag{30}$$

$$= Q_{\boldsymbol{\pi}}(\boldsymbol{\pi};\hat{\lambda}) + Q_{\mathbf{A}}(\mathbf{A};\hat{\lambda}) + Q_{\theta}(\theta;\hat{\lambda}) \tag{31}$$

where $Q(\lambda;\hat{\lambda})$ in (31) is decomposed as the sum of 3 parameter-specific Q functions corresponding to $\boldsymbol{\pi}$, $\mathbf{A}$ and θ, which can be optimized independently of each other. The expressions for $Q_{\boldsymbol{\pi}}(\boldsymbol{\pi};\hat{\lambda})$, $Q_{\mathbf{A}}(\mathbf{A};\hat{\lambda})$ and $Q_{\theta}(\theta;\hat{\lambda})$ are easily obtained from (30) and (31). We seek to find optimal positive-valued parameters $\boldsymbol{\pi}$ and $\mathbf{A}$ to maximize the functions $Q_{\boldsymbol{\pi}}(\boldsymbol{\pi},\ \hat{\lambda})$ and $Q_{\mathbf{A}}(\mathbf{A};\hat{\lambda})$, respectively, subject to the constraints $\sum_{i=1}^{L} \boldsymbol{\pi}_i = 1$ and $\sum_{j=1}^{L} a_{ij} = 1, \forall_i$, also in respective order. We can use Lagrange multipliers to obtain the following update relations for $\boldsymbol{\pi}$ and $\mathbf{A}$ in terms of $\hat{\lambda}$

$$\boldsymbol{\pi}_i = \frac{P(\mathbf{X}, w_0 = i;\hat{\lambda})}{P(\mathbf{X};\hat{\lambda})}, \tag{32}$$

$$a_{ij} = \frac{\sum_{t=1}^{N} P(\mathbf{X}, w_t = i, w_{t+1} = j;\hat{\lambda})}{\sum_{t=1}^{N} P(\mathbf{X}, w_t = i;\hat{\lambda})}. \tag{33}$$

The Q function $Q_{\theta}(\theta;\hat{\lambda})$ for the Gaussian mixture parameters $\theta_i = \{c_{i,j=1}^{K}, \boldsymbol{\mu}_{i,j=1}^{K}, \boldsymbol{\Sigma}_{i,j=1}^{K}\}$ can be expressed as $Q_{\theta}(\theta;\hat{\lambda}) = Q_{\mathbf{c}}(\mathbf{c};\hat{\lambda}) + Q_{\mathbf{b}}(\boldsymbol{\mu},\boldsymbol{\Sigma};\hat{\lambda})$. The update relation for the Gaussian mixture probabilities $\mathbf{c}$ is found using Lagrange multipliers to maximize $Q_{\mathbf{c}}(\mathbf{c};\hat{\lambda})$ subject to the constraint $\sum_{j=1}^{K} c_{ij} = 1$; the result is given below

$$c_{ij} = \frac{\sum_{t=1}^{N} P(w_t = i, m_t = j|\mathbf{x}_t;\hat{\lambda})}{\sum_{t=1}^{N}\sum_{m=1}^{K} P(w_t = i, m_t = m|\mathbf{x}_t;\hat{\lambda})}, \tag{34}$$

where $P(w_t = i, m_t = j|\mathbf{x}_t;\hat{\lambda})$ is the so called "responsibility probability" [5] of the joint event of model state $w_t = i$ and mixture component $m_t = j$ given the t^{th} observation vector $\mathbf{x}_t$. The update expressions for the means $\boldsymbol{\mu}$ and covariance matrices $\boldsymbol{\Sigma}$ are found by setting the derivatives $\frac{\partial Q_{\mathbf{b}}(\boldsymbol{\mu},\boldsymbol{\Sigma};\hat{\lambda})}{\partial \boldsymbol{\mu}_{ij}}$ and $\frac{\partial Q_{\mathbf{b}}(\boldsymbol{\mu},\boldsymbol{\Sigma};\hat{\lambda})}{\partial \boldsymbol{\Sigma}_{ij}}$ to zero. The update expressions are given in (35) and (36) below.

$$\boldsymbol{\mu}_{ij} = \frac{\sum_{t=1}^{N} P(w_t = i, m_t = j|\mathbf{X}; \hat{\lambda}) \cdot \mathbf{x}_t}{\sum_{t=1}^{N} P(w_t = i, m_t = m|\mathbf{X}; \hat{\lambda})}, \tag{35}$$

$$\boldsymbol{\Sigma}_{ij} = \frac{\sum_{t=1}^{N} \left[P(w_t = i, m_t = j|\mathbf{X}; \hat{\lambda})\right] (\mathbf{x}_t - \boldsymbol{\mu}_{ij})(\mathbf{x}_t - \boldsymbol{\mu}_{ij})^T}{\sum_{t=1}^{N} P(w_t = i, m_t = m|\mathbf{X}; \hat{\lambda})}. \tag{36}$$

2.3 Neural Decoding in Brain-Computer Interfaces

Systems involving electrodes implanted into the cortex of an animal subject are called intracortical brain-computer interface (BCI) systems. Intracortical BCIs are valuable tools for studying brain activity in humans and animals. For persons with impairments to hearing, sight, motor function or communication, *neural prosthetic systems* based on intracortical BCIs offer great potential for improving quality of life.

Some notable studies in intracortical BCIs include [43, 78, 86, 10]. In all of these studies, a microwire electrode (usually an array of electrodes) was implanted deep into a carefully chosen region of the brain to trace the extracellular electrical activity of a population of neurons. The electrical trace is then *decoded* in some way to carry out the task. In [44], for example, extracellular electrical signals in motor cortex were decoded and used to control a "virtual hand" on a computer screen for a human subject living with Locked-In Syndrome.

The dominant approach to decoding in intracortical BCIs is to detect **neuronal action potential events** in the extracellular electric trace. It is widely accepted that the information content can be decoded from the *rate* at which neuronal action potentials or "firing" events occur [8]. As a result, given K neurons in the vicinity of an electrode, it is necessary to determine the occurrence times of all action potential events so that an estimate of the firing rate can be computed for each neuron. Statistical pattern recognition or other approaches are then used to decode extracellular neural signals into the intended result.

In the remainder of this section, we briefly review the mechanism behind the

action potential event within neuronal cells. We then review the state of statistical methods for the task of spike-sorting, which is to identify neuronal action potential events in an extracellular trace and to discriminate between neurons based on the set of observed waveforms. Finally, we review firing rate modeling in neural spike trains using Poisson point processes.

2.3.1 The Neuronal Action Potential

Much of human brain function is accomplished by the activity of billions of neurons composed into large electrical communication networks in the brain. Short, pulse-like variations in the electric potential across the neuronal cell membrane called *action potentials*, are transmitted between neurons, and form the basis for communication in neuronal networks. Action potentials are the result of ionic currents, i.e., the controlled passage of specific ions in and out of the cell membrane. During an action potential event, a distinctive waveform shape is observed in the electric potential across the membrane. In the remainder of this section, we give a brief description of the mechanism behind the neuronal action potential.

The action potential event in a neuron involves the passage of two types of cations, sodium (Na+) and potassium (K+), through its cell membrane. For each type of ion, the cell membrane contains voltage-sensitive channels that open and close to permit or inhibit its *passive* permeation either in or out of the cell. Also found in the cell membrane are special protein complexes called "ionic pumps," which *actively* eject Na+ ions from the cell and infuse K+ ions into the cell from outside. In the equilibrium or "resting" state of a neuron, the ionic pumps help maintain a constant molar concentration ratio of ions outside and inside the cell for both Na+ and K+ ions. The $outside : inside$ concentration ratios for K+ ions and Na+ ions are roughly 20mM : 400mM and 440mM : 50mM, respectively. As a result, when K+ channels open, K+ ions tend to leave the cell by diffusion; conversely, Na+ ions *enter* the cell

when the Na+ channels open. Also, with the concentration ratios kept more or less constant, the electric potential across the membrane (which can be computed using the well-known Nernst equation [95]) remains constant as well at about -70mV [1]; this is called the "resting potential." At the resting potential, the membrane is said to be polarized.

The permeability of the cell membrane to Na+ ions is a direct function of the portion of voltage-sensitive Na+ channels that are in the "open" state at any given moment. The portion of open channels itself is a function of the electric potential inside the cell, and when an electrical stimulus is applied to even a small patch of membrane, Na+ channels in the vicinity begin to open. As Na+ cations diffuse into the cell, through these newly opened channels, the electric potential increases further, thus causing more Na+ channels to open. A plot of the electric potential across the cell membrane due to ion flow for a stereotypical action potential event is given in the upper panel of Figure 1. The start of the **action potential** is marked by the moment the potential inside the cell crosses a critical threshold ($\approx$ -60mV). At this point, the number of open Na+ channels increases dramatically, causing a rapid influx of Na+ and a concomitant rapid increase in the electric potential; this event is indicated in Figure 1 with the number "1." When the potential reaches a certain level ($\approx$ 40 mV) two near-simultaneous events, indicated with the number "2" in Figure 1, cause it to reach a peak and then fall quickly: the Na+ channels transition to an "inactivated" state, abruptly halting the influx of Na+, and the K+ channels enter their "open" state, causing K+ ions to diffuse out of the cell, lowering the potential. The efflux of K+ ions can actually cause the potential to fall below the resting potential. At this point, the K+ channels close (event "3" in Figure 1), and the Na+ and K+ ionic pumps restore the *outside* : *inside* molar concentrations of Na+ and K+ to their resting levels, thus also restoring the resting potential.

[1] By convention, the potential is measured inside the cell with respect to outside.

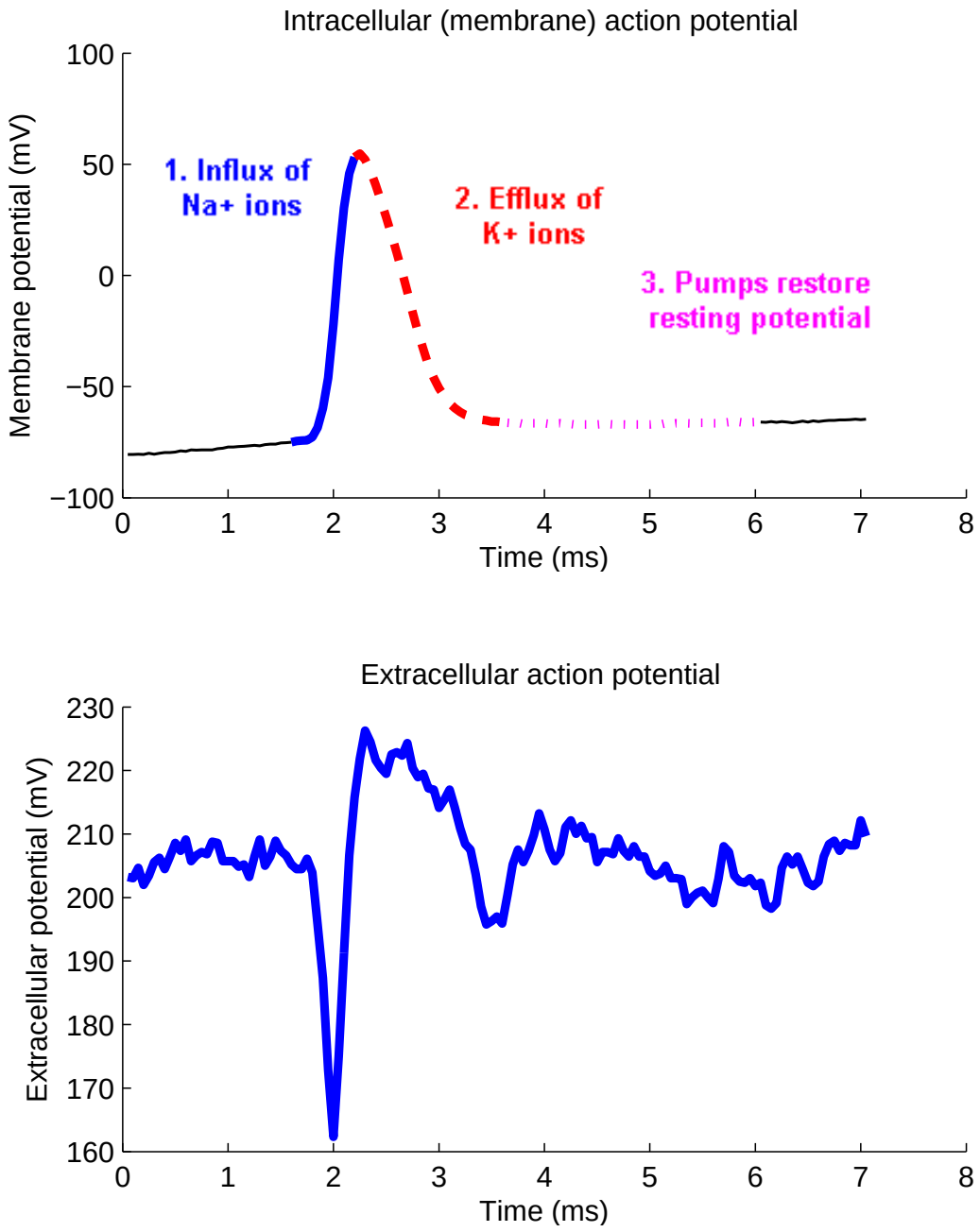


Figure 1: Action potential ion flow. Upper panel: the membrane potential of a neuronal cell during an action potential event measured with an intracellular (IC) electrode. Lower panel: simultaneously recorded trace using an extracellular (EC) electrode.

2.3.1.1 Observing action potentials with electrodes in BCIs

To observe action potentials from a single neuron of interest, it is possible, although difficult, to use an **intracellular (IC) electrode**, i.e., an electrode placed inside the neuronal cell membrane with the ground potential outside. While the IC electrode allows for near-perfect detection of action potential events for a single neuron, it is extremely difficult to place in multiple neurons and nearly impossible to maintain in an awake animal subject at all. Instead, most intracortical BCIs use electrodes to observe electric potentials in the close vicinity (i.e., within a few microns) of one or more cortical neurons; these are called **extracellular (EC) electrodes**.

An extracellular electrode placed in the vicinity of a population of neurons can be used to observe transient ionic currents due to action potential events. An action potential from a nearby neuron is typically observed as a "spike" in an extracellular trace. The shape of the observed extracellular spike waveform is very likely to differ from the corresponding intracellular action potential waveform. This is mainly due to the morphology, polarity and position of the EC electrode with respect to the cell membrane. This is illustrated in Figure 1, where simultaneously recorded intracellular and extracellular traces are given for a single action potential event in the upper and lower panels, respectively. This same idea makes it possible to use a single electrode to observe action potential events for multiple neurons, and to discriminate between distinct neurons with (possibly) distinct waveform shapes using signal processing methods. This operation is commonly called "spike-sorting" and is discussed in greater detail in Section 2.3.2 .

2.3.2 Spike Sorting

Traces of multi-channel extracellular electric potentials are valuable for studying the behavior of neurons and have far-reaching implications for motor and communication prostheses through brain-computer interfaces. A cortically implanted electrode

records the superimposed electrical activity of a population of neurons. Most of these systems, however, rely on isolating the activity of individual neurons (referred to as "single unit activity") and small clusters of neurons (multi-unit activity). As a result, significant further signal processing operations are necessary.

Given a trace of extracellular electric potentials, spike-sorting is the task of identifying the neuronal sources of action potential "spikes" in the signal. In most cases, the end result of the spike sorting task is the identity and firing times for a population of neurons or neuronal clusters. Most approaches to spike sorting involve two high-level operations: (1) detecting the occurrence times of all threshold-crossing spikes in the waveform and (2) assigning each spike to a neuronal source. In this section, we briefly review commonly used approaches to the detection and classification stages for the spike-sorting task.

2.3.2.1 Detection

Wideband traces of extracellular potentials on microwire electrodes are usually sampled between 20 kHz and 40 kHz. Typically, a bandpass filter is applied to emphasize neuronal firing activity in a frequency range of 100 Hz to 10000 Hz, and to eliminate low frequency waveform drift and unwanted high frequency noise. Specifications for pre-emphasis operations can vary widely depending on the electrode, the designer's preference and other factors.

After pre-emphasis operations are applied, spikes in an extracellular trace are typically detected as peaks crossing some magnitude threshold. As described in Section 2.3.1, the observed voltage waveform for a neural action potential rises to a peak as corresponding with a rapid influx of Na+ ions. If the magnitude of the action potential peak is significantly greater than the background noise level, action potential waveforms can be reliably extracted by setting a voltage threshold. Typically, the waveform is defined as a short duration before and after the peak on the order

of 1 to 2 ms. Spike detection is complete when all of the detected action potential waveforms in a given trace are collected into a data set. Note that while there are many methods for spike detection [72, 63, 45], the target result is the same, i.e., a collection of extracted spike waveforms.

2.3.2.2 Classification (Clustering)

The result of the detection operation is a set of collected time-domain spike waveforms, each of the same length. We would like to classify these waveforms according to which neurons produced them. However, since the action potential mechanism is typically not directly observed, classification in neural spike sorting is an unsupervised pattern recognition or clustering task.

Although research into automatic methods for spike sorting has a long history, the most common methods in practice for identifying neuronal units in an extracellular trace are performed by hand. In so called **cluster cutting** methods, an expert, with the assistance of a computer, visually identifies features in the set of spike waveforms such as the height of the peak due to the Na+ influx or the depth of the valley due to the K+ efflux (cf. Section 2.3.1). The sorter manually marks a region in feature space for each neuron cluster using a polygon or other convex hull shape, based on his or her own expertise and intuition [51].

Finally, *automatic* methods for spike sorting use rigorous signal processing and statistical modeling approaches to the spike classification or clustering problem. Perhaps the most immediate advantage of these methods over cluster cutting is a significant reduction in effort. This is especially important for large microelectrode arrays which can have dozens of individual electrodes, making manual cluster cutting prohibitively cumbersome.

For most automatic spike sorting methods, feature extraction and dimensionality reduction are applied to the set of extracted time-domain waveforms after detection.

Given a set of N waveforms, the result of this step is a matrix $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ where each parameterized waveform $\mathbf{x}_i$ is of dimension D, and D is considerably smaller than the length (in samples) of the original time-domain waveform. Principal components analysis (PCA) is very common method for waveform parameterization [51] along with several other techniques based on the discrete wavelet transform (DWT) [72, 66].

2.3.2.3 Statistical Methods for Clustering

While many methods and frameworks for automatic spike sorting have been proposed in the literature, in this work, we focus on probabilistic approaches to the problem. Particularly, we focus on clustering approaches based on the Expectation-Maximization (EM) algorithm. In Sections 2.2.1 and 2.2.3 we discuss the EM algorithm for Gaussian mixture models and hidden Markov models, respectively, as applied to the task of density estimation. With a few basic assumptions, the same approach can be applied to clustering neural waveforms.

It is a well-known result that, after applying principal components analysis (PCA) to a set of action potential waveforms, the first several principal components are well described by a multivariate Gaussian distribution [51]. Given K neurons in the vicinity of an intracortically implanted electrode, the expression for the likelihood of a parameterized action potential waveform $\mathbf{x}$, for the k^{th} neuron is as given below

$$p(\mathbf{x}|C_k) = p(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{D/2} \left|\boldsymbol{\Sigma}_k\right|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k (\mathbf{x} - \boldsymbol{\mu}_k)\right\}, \quad (37)$$

where $\mathbf{x}$ is of dimension D and C_k is a discrete-valued variable indicating the k^{th} neuron. Given a set of N action potential waveforms, the likelihood for the dataset $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ is given by

$$P(\mathbf{X};\theta) = \prod_{i=1}^{N} p(\mathbf{x}_i;\theta) = \prod_{i=1}^{N}\sum_{k=1}^{K} p(\mathbf{x}_i|C_k)P(C_k) \tag{38}$$

$$= \prod_{i=1}^{N}\sum_{k=1}^{K} p(\mathbf{x};\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\pi_k, \tag{39}$$

where $\pi_k = P(C_k)$, and the parameter set θ is defined $\theta = \{\pi_{k=1}^K, \boldsymbol{\mu}_{k=1}^K, \boldsymbol{\Sigma}_{k=1}^K\}$. Since action potential waveforms in $\mathbf{X}$ are observed only as spikes on an extracellular trace, the variable $\{C_i\}_{i=1}^N$, which indicates which neuron produced each spike, cannot be known with certainty.

Note that the expression for the likelihood $P(\mathbf{X};\theta)$ in (38) is equivalent to the expression for the log-likelihood in (11) if we simply substitute the neuronal class variable C_k for the Gaussian mixture component variable m_j and take the log of $P(\mathbf{X};\theta)$. Though the purposes of the clustering problem in (38) and the density estimation problem in (11) are different, the EM solution, derived in detail in Section 2.2.1 is identical.

2.3.3 Neural Spike Trains and Firing Rates

Approaches to the problem of neural decoding are typically based on characterizing the neural response to some stimulus or activity of interest. The neural response is comprised of a sequence of action potentials or "spike" events, with information about the stimulus or activity of interest encoded primarily in the timing of the events. For this reason, we represent the neural response as an impulse train $\rho(t)$, often called a "spike train," such that

$$\rho(t) = \sum_{i=1}^{n} \delta(t - t_i). \tag{40}$$

It is useful, then, to model the neural response as a stochastic process consisting of firing times t_i for $i = 1, 2, \ldots, n$. The random process representation for spike trains provides a rigorous, extensible framework for modeling the spike data themselves, as

well as any important conditions of the data including the stimulus of interest. For stimulus modeling especially, we will generally assume that the spike train response depends on some notion of a time rate or "firing rate" $r(t)$ of neuronal action potential events. The firing rate can be constant or time-varying. If we assume a constant firing rate r, the value of r is simply the average firing rate of n spikes over an interval of length T as given below

$$r = \frac{n}{T} = \frac{1}{T}\int_0^T \rho(\tau)d\tau. \tag{41}$$

More generally, we can define a time-varying function $r(t)$ to be the *instantaneous* firing rate. The quantity $r(t)$ should measure or commensurate the ratio of spikes fired in a very short interval Δt to the length of the interval. The time-varying firing rate $r(t)$ can be defined as follows

$$r(t) = \frac{1}{\Delta t}\int_t^{t+\Delta t} \langle \rho(t) \rangle d\tau, \tag{42}$$

where the expectation $\langle \rho(t) \rangle$ is the neural response function averaged over many time-aligned trials with the same stimulus applied each time. As Δt in (42) approaches 0, the firing rate $r(t)$ approaches a smooth, continuously varying function [6, 19].

In characterizing the neural response to a stimulus, we model the spike train in $\rho(t)$ as generated by an underlying stochastic process that depends, in some way, on the firing rate function $r(t)$. If we make the assumption that each spike time t_i, is completely independent of all other spike times, then the spike train $\rho(t)$ is a **Poisson random process**, completely characterized by the firing rate. Methods of decoding single neural spike trains, as well as ensembles of neural spike trains common to intracortical brain-computer interfaces, rely heavily on the Poisson process model. The Poisson model of spike generation simplifies neural decoding, since it is only necessary to compute an estimate of the firing rate of observed spike trains to make inferences about neural data.

We will discuss two varieties of Poisson processes important for modeling neural responses: the **homogeneous Poisson process** for which the firing rate r is constant, and the **inhomogeneous Poisson process** for which a time-varying firing rate $r(t)$ is assumed.

2.3.3.1 Homogeneous Poisson Processes

Assuming a constant firing rate r, the number of independent events occurring in a time interval of length T is a Poisson random variable. The probability $P_T[n]$ of n events occurring in the interval is given by

$$P_T[n] = \frac{(rT)^n}{n!} \exp(-rT). \tag{43}$$

A point process of n time occurrences $t_1, t_2, \ldots, t_n$ in the same interval is called a Poisson point process and its probability $P[t_1, t_2, \ldots, t_n]$ is given by

$$P[t_1, t_2, \ldots, t_n] = n! P_T[n] \left(\frac{\Delta t}{T}\right)^n, \tag{44}$$

where Δt is the length of a very small time interval around each event t_i. Substituting the expression for $P_T[n]$ into (44), we obtain

$$P[t_1, t_2, \ldots, t_n] = (rT)^n \exp(-rT) \left(\frac{\Delta t}{T}\right)^n \tag{45}$$

$$P[t_1, t_2, \ldots, t_n] = \exp(-rT)(r\Delta t)^n \tag{46}$$

$$p[t_1, t_2, \ldots, t_n] = \exp(-rT) r^n = P[t_1, t_2, \ldots, t_n]/(\Delta t)^n, \tag{47}$$

where, for each t_i, $p[t_i]$ (with a lowercase "p") is the probability density function. Since Δt is a small time interval, $p[t_i]\Delta t$ is a near approximation to the area under the probability density curve at time t_i and, in turn, the probability $P[t_i]$. For n time occurrences, $p[t_1, t_2, \ldots, t_n](\Delta t)^n = P[t_1, t_2, \ldots, t_n]$ as in (47). Since its firing rate r is constant, $p[t_1, t_2, \ldots, t_n]$ is said to be a homogeneous Poisson point process.

2.3.3.2 Inhomogeneous Poisson Processes

In the more general case where the instantaneous firing rate $r(t)$ varies with time, we can still model with the assumption that all spike occurrence times are independent of each other. This is called an inhomogeneous Poisson process. As with the homogeneous case, the independent spike assumption implies that the process is completely characterized by the firing rate. The expression for the probability density function for an inhomogeneous Poisson process is given below

$$p[t_1, t_2, \ldots, t_n] = \exp\left(-\int_0^T r(t)dt\right) \prod_{i=1}^{n} r(t_i) \tag{48}$$

Note that substituting $r(t) = r$ into the expression above, we obtain the expression for the homogeneous case in (47).

2.3.3.3 Limitations of Poisson Processes for Neural Spike Trains

Poisson process models are completely characterized by their firing rate parameter and are especially useful for decoding neural spike trains. As a model of the neural spike train itself, the Poisson point process has some important limitations. For example, following every neural firing is a short **refractory period** during which it is impossible for another firing event to occur. The independent spike assumption does not hold well since every spike has, at the very least, an implicit dependency on the previous firing. Renewal processes, which explicitly model dependencies between consecutive time occurrences are useful for when more detailed point process models are required.

CHAPTER III

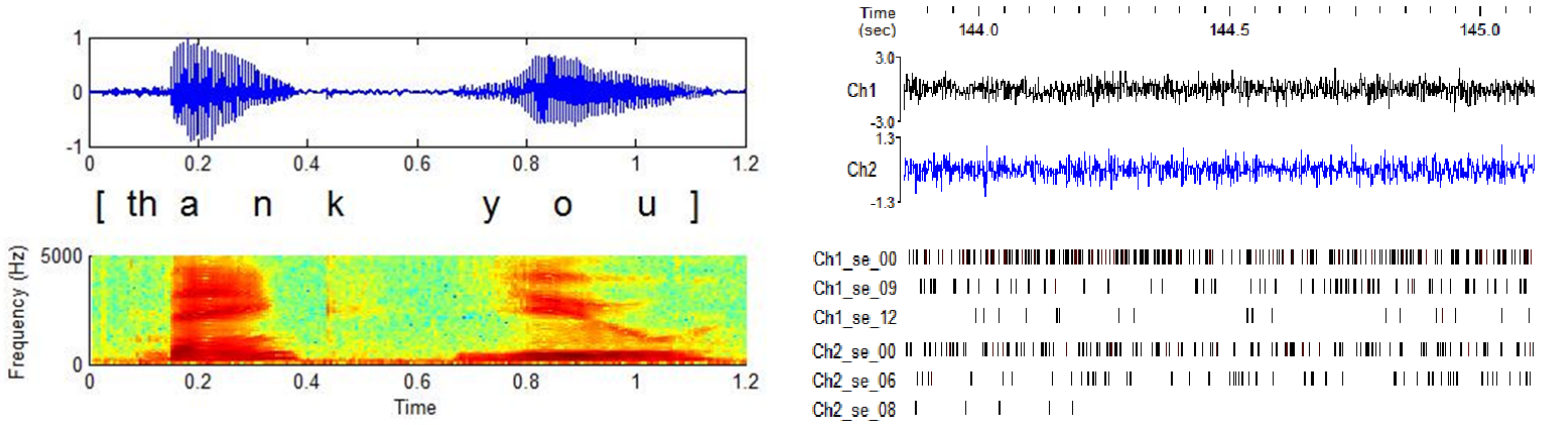
DETECTION-BASED AUTOMATIC SPEECH RECOGNITION

In this chapter we investigate detecting articulatory speech attributes in speech audio signals using support vector machine (SVM) classifiers, and combining these and other classifiers for automatic speech recognition. We discuss the selection of articulatory attributes for detection, training SVM detectors for each attribute and we report the detection results obtained. We then report the results of combining attribute detection scores for continuous phone recognition.

3.1 Motivation

The decoding operation in a neural speech prosthesis has many important structural similarities to the well-known automatic speech recognition (ASR) problem. Figure 2 (a) gives a plot of a speech audio signal, "Thank you" and its spectrogram, i.e., a depiction of its short-time discrete Fourier transform (STFT). In nearly all ASR systems, the input signal is some multivariate, frequency-based representation of a speech audio signal. The input to the decoding stage of a neural prosthesis, as we define it, is an ensemble of action potential firing times or instantaneous firing rate estimates from a population of speech-related cortical neurons; a raster plot of neuronal firing times is given in Figure 2 (b). For both of these problems, the objective is to decode a multivariate, continuous-valued input signal into a time-aligned sequence of phonemic or linguistic tokens or whole words.

As we will show in Chapters 4 and 5, our approach to the neural speech prosthesis problem, which builds strongly on research in ASR, is to decode neurological signals



(a) Speech audio signal

(b) Neural signal

Figure 2: Multivariate parameterizations of speech signals. (a) Speech audio signal. (b) Neural signal from speech motor cortex.

specifically related to speech motor function, such as the movement of the lips, tongue, jaw, glottis, etc., collectively referred to as the set of *speech articulators*. In this chapter, we motivate the speech motor approach for our neural speech prosthesis with an *articulatory* approach to automatic speech recognition for speech audio signals. We train statistical models of various phonological attributes of speech signals related to the articulators used, and the manner (e.g., *with* frication, plosion, or sonority) and place (e.g., *at* the velum or the alveolar ridge) of articulation. Our goal is to show that explicit modeling of articulatory information can be used to decode speech content.

3.2 Detection-Based Automatic Speech Recognition

The most commonly adopted approach to the task of automatic speech recognition is to train acoustic models for a prescribed alphabet of short linguistic units, usually at the subword level, and to use dynamic programming methods to find the best sequence of words for a given spoken utterance. While much of the success in the performance of ASR systems is directly attributable to this paradigm and its variants, a wide body of expert knowledge in linguistics, acoustic phonetics and phonology is largely unused in modern ASR Systems.

In this chapter we discuss a detection-based paradigm for ASR, proposed to address some of the limitations of modern ASR systems and to narrow the significant performance gap between ASR and human speech recognition. Specifically, we present methods of detector design in the Automatic Speech Attribute Transcription (ASAT) project, where we have incorporated detectors of various attributes of the speech signal (sometimes referred to in the literature as *distinctive features* or *phonological features* or *acoustic-phonetic features*) into our approach to ASR [7].

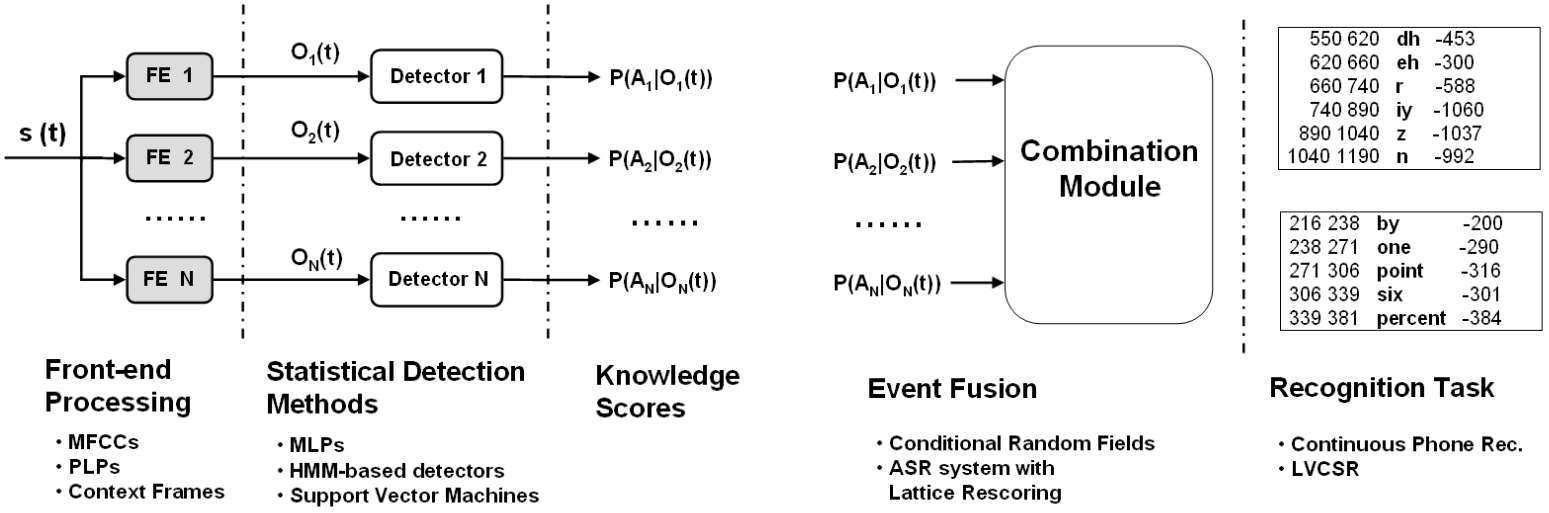
Figure 3 illustrates the detection-based ASR paradigm. At the front end is a bank of detectors of useful and meaningful attributes of the speech signal. The outputs of

these detectors, typically confidence scores for each attribute, are fused to infer higher-level evidences for the speech recognition task. Our selection of speech attributes is based directly on articulatory modeling and acoustic phonetics. Detection-based ASR then represents an opportunity to effectively and methodically incorporate expert knowledge of speech production into speech recognition systems.

The front end of the ASAT detection-based ASR system is depicted in Figure 3 (a). The speech signal is first analyzed by a bank of detectors, each producing a confidence score or posterior probability pertaining to some acoustic-phonetic attribute. The design of these detectors, the optimization of their parameters and the selection of the set of attributes to detect are all critical design problems for the detection-based ASR paradigm. In Section 3.3, we discuss our contribution to the ASAT project, i.e., using support vector machine classsifiers to detect phonological attributes of the speech signal. In Section 3.4, we briefly discuss the full ensemble of attribute detectors developed by our collaborators, and the methods and results of combining them for automatic speech recognition. All experiments were performed on the TIMIT speech database [21].

3.3 SVM-based Attribute Detectors

Support vector machine (SVM) classifiers belong to the kernel machines family of pattern recognition methods. SVMs are the most widely used of the kernel machine methods and have been extensively applied to many pattern recognition problems, including speech recognition [24, 37]. Let $\mathbf{x}$ and $\mathbf{y}$ be two n-dimensional data points. For kernel machine methods, points in the n-dimensional input space $\mathbb{R}^n$ are implicitly mapped to a high-dimensional feature space $\mathbb{R}^{n_K}$ using a kernel function. According to Mercer's condition, the inner product of two vectors $\boldsymbol{\phi}(\mathbf{x})$ and $\boldsymbol{\phi}(\mathbf{y})$, in a high-dimensional feature space, can be computed with $K(\mathbf{x}, \mathbf{y}) = \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{y}) \rangle$, where K is a kernel function and $\boldsymbol{\phi} : \mathbb{R}^n \rightarrow \mathbb{R}^{n_k}$ is the nonlinear function mapping the



(a) Front end.

(b) Decoding.

Figure 3: ASAT Detection-Based ASR.

input space to the feature space. Examples of kernel functions include a linear kernel $K(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T\mathbf{y}$ and a radial basis function (RBF) kernel $K(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/\sigma^2}$. Kernel machine methods can then use linear classification techniques involving inner products in a non-linear space, achieving enhanced performance in many classification tasks.

Support Vector Machines find the optimal separating hyperplane by finding a vector $\hat{\boldsymbol{\alpha}}$ which maximizes the expression in (49)

$$\hat{\alpha} = \arg\max_{\boldsymbol{\alpha}} \sum_{i=1}^{N} -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{49}$$

subject to

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{50}$$

$$0 \leqslant \alpha_i \leqslant C, \ \forall_i \tag{51}$$

where the parameter C controls the trade-off between training error and generalization.

For SVMs, class-conditional probabilities are approximated using a parametric approach [68] in which a sigmoid function of the form

$$P(y = +1|\mathbf{x}) = 1/(1 + e^{Af(\mathbf{x})+B}) \tag{52}$$

maps SVM projections to class conditional probabilities. The parameters A and B are determined from cross-validation.

3.3.1 Articulatory Features

In this section, we discuss the selection of articulatory attributes for detection in speech signals. Articulatory gestures used for speech production can be roughly arranged along 3 broad categories, i.e., *voicing*, *manner of articulation* and *place of articulation* [46]. The set of ARPABET English phonemes (including silences) used in the TIMIT database is given in Table 1, organized according to these three categories.

Voicing is a binary variable indicating the activity of the glottis. The place category indicates the location in the vocal tract at which air flow is constricted to make a sound, and the manner category describes nature of the constriction. Among manner attributes, air flow is constricted the least with vowels, while stops involve a complete obstruction of air followed by a plosion. Also, for vowel sounds, an indicator of lip-rounding ("rnd+" or "rnd-") is given in Table 1. Including silence, there a total of 17 manner and place attributes listed in Table 1. Adding "roundminus," "roundplus," voicedminus," and "voicedplus" we obtain a set of 21. We train 2-class SVM classifiers for each of these articulatory speech attributes.

3.3.2 SVM Detection Experiments

Using the 21 speech attributes described in Section 3.3.1, we trained support vector clasifiers (1 for each attribute) on the phonetically labeled TIMIT database. We use Mel-frequency cepstral coefficients (MFCCs) for front-end feature extraction, with 12 cepstral + 1 energy coefficient along with 1st and 2nd derivative coefficients (called Δ's and $\Delta\Delta$'s) for a 39-length feature vector.

We train a 2-class SVM, where the two classes are data corresponding to each attribute and its complement. We use a radial basis function (RBF) kernel, and estimate the posterior probability for each attribute using a sigmoid mapping of the SVM projection as in (52). The posterior probability estimates are then used as attribute scores.

We train the classifiers on a relatively small (i.e., 50 utterances), randomly selected subset of the TIMIT training data set due to memory limitations. We then compute attribute scores for the entire TIMIT test set of 1344 utterances. We evaluate performance for attribute detection using receiver operating characteristics (ROC) curves, which illustrate the trade off between Type I or "miss" errors and Type II or "false

Table 1: ARPAbet phone set and articulatory attributes.

Phone	Manner	Place	Voice	Phone	Manner	Place	Voice
aa	vowel (rnd+)	low	Y	iy	vowel (rnd+)	high	Y
ae	vowel (rnd+)	low	Y	jh	fricative	high	Y
ah	vowel (rnd+)	mid	Y	k	stop	velar	N
ao	vowel (rnd+)	high	Y	kcl	stop	velar	N
aw	vowel (rnd+)	low	Y	l	lateral	coronal	Y
ax	vowel (rnd-)	back	Y	m	nasal	labial	Y
axr	vowel (rnd+)	mid	Y	n	nasal	coronal	Y
ay	vowel (rnd+)	low	Y	ng	nasal	velar	Y
b	stop	labial	Y	nx	approximant	coronal	Y
bcl	stop	labial	Y	ow	vowel (rnd+)	mid	Y
ch	fricative	high	N	oy	vowel (rnd-)	low	Y
d	stop	coronal	Y	p	stop	labial	N
dcl	stop	coronal	Y	pcl	stop	labial	N
dh	fricative	dental	Y	q	stop	glottal	N
dx	stop	coronal	Y	qcl	stop	glottal	N
eh	vowel (rnd+)	mid	Y	r	approximant	retroflex	Y
el	lateral	coronal	Y	s	fricative	coronal	N
em	nasal	labial	Y	sh	fricative	high	N
en	nasal	coronal	Y	t	stop	coronal	N
er	vowel (rnd-)	retroflex	Y	tcl	stop	coronal	N
ey	vowel (rnd+)	mid	Y	th	fricative	dental	N
f	fricative	labial	N	uh	vowel (rnd-)	high	Y
g	stop	velar	Y	uw	vowel (rnd+)	high	Y
gcl	stop	velar	Y	v	fricative	labial	Y
hh	fricative	glottal	N	w	approximant	labial	Y
hv	fricative	glottal	N	y	approximant	high	Y
ih	vowel (rnd+)	high	Y	z	fricative	coronal	Y
ix	vowel (rnd+)	front	Y	zh	fricative	high	Y
h#	silence			pau	silence		
#h	silence			epi	silence		

alarm" errors. ROC curves for manner of articulation and place of articulation attributes, as well as silence, voicing and lip-rounding are given in Figure 4. Generally, the best performance is obtained for manner of articulation attributes and the silence and voicing attributes.

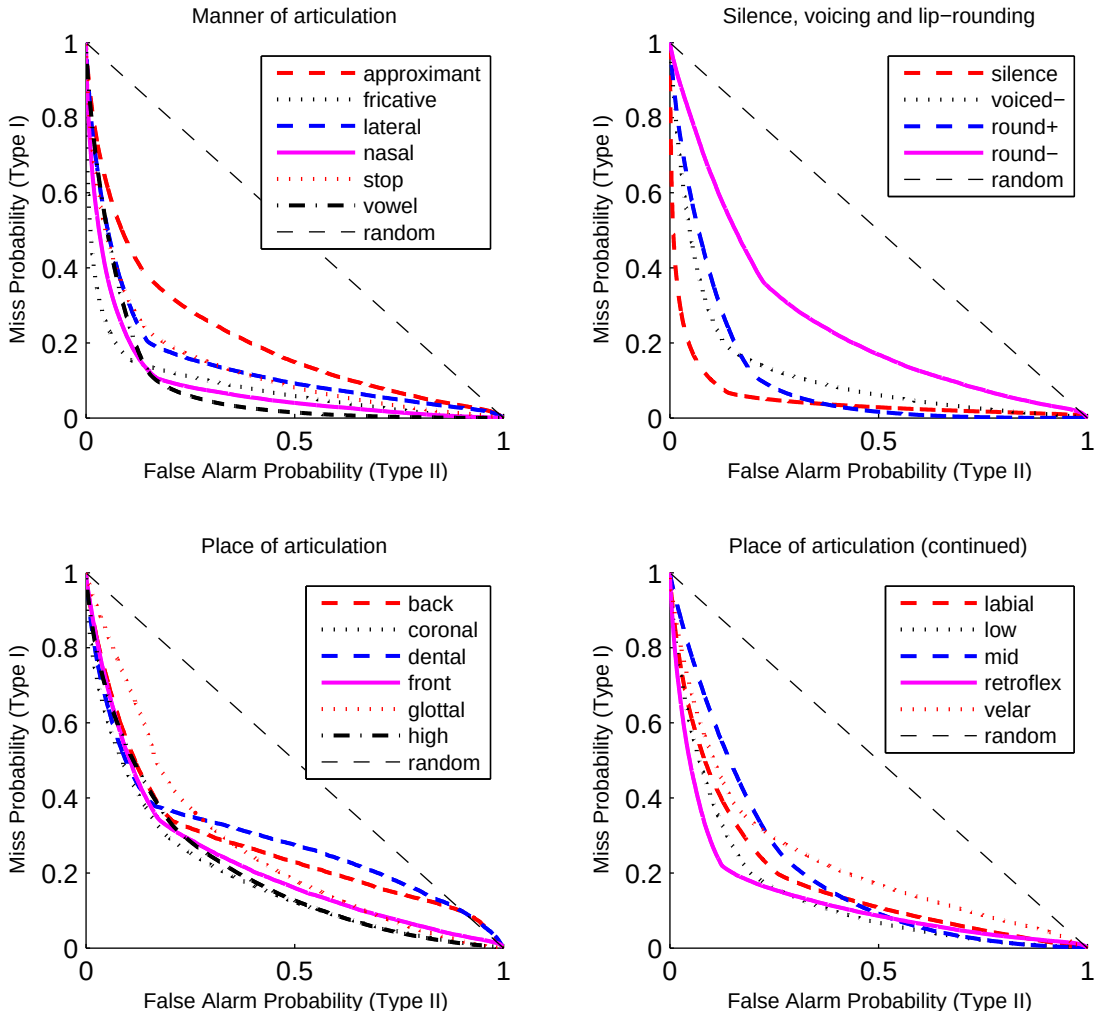


Figure 4: ROC curves for SVM detection of articulatory attributes.

3.4 Ensemble of Speech Attribute Detectors

As illustrated in Figure 3 (b), the "Event Fusion" stage involves combining an ensemble of speech attribute detectors to produce a time-aligned recognition result. In this section we discuss the full set of attribute detectors (in addition to the SVM detectors discussed earlier) used in the ASAT project and methods for combining attribute detectors to achieve the recognition result.

3.4.1 Other Speech Attribute Detectors

Detectors of varying design methodologies and front-end processing techniques each have their own strengths and advantages and can be easily incorporated into our framework. In addition to our support vector machine (SVM) classifiers, multi-layer perceptrons (MLPs) [34], hidden Markov models, and time-delay neural nets (TDNNs) were developed by our collaborators to detect various acoustic-phonetic distinctive attributes of the speech signal, and to detect boundaries between phones and phonological features [89] over the course of the ASAT project. The full set of detectors and attributes used in the ASAT project is given in Table 2. In this section, we briefly describe speech attribute detectors developed by our collaborators.

Table 2: Summary of detectors, front-end processing methods and speech attributes.

Methods of Detection	Front-end Processing	Speech Attributes
MLP (SPE)	13 MFCCs 10 msec frames	**SPE Classes:** vocalic consonantal high back low anterior coronal round tense voice continuant nasal strident silence (14 attributes)
SVM	13 MFCCs 9 context frames 10 msec frames	coronal dent fricative glottal high labial low mid nasal roundminus roundplus silence stop velar voicedminus voicedplus vowel (17 attributes)
HMM-Based	13 MFCCs + Δ + ΔΔ 10 msec frames	
Multi-class MLPs	13 PLPs + Δ + ΔΔ 9 context frames 10 msec frames	**Sonority**: Obstruent Silence Sonorant Syllabic Vowel **Voicing**: NA Voiced Voiceless **Manner**: Approximant Flap Fricative NA Nasal NasalFlap Stop-Closure Stop **Place**: Alveolar Dental Glottal Labial Lateral NA Palatal Rhotic Velar **Height**: High Low-High Low Mid-High Mid NA **Backness**: Back Back-Front Central Front NA **Roundness**: NA NonRound NonRound-Round Round-NonRound Round **Tenseness**: Lax NA Tense (44 attributes)

3.4.1.1 MLP-based detectors for Sound Pattern of English Classes

Using the Sound Pattern of English (SPE) features defined by Chomsky and Halle [13] as speech attributes, we built and optimized a set of Multi-Layer Perceptrons to detect each of the 14 binary-valued SPE features. The 61 TIMIT phonemes are mapped to the 14 SPE features, and the detection is done on each utterance frame

by frame. We tested this architecture using both 2-layer and 3-layer MLPs using the Netlab and Matlab toolboxes. The input layer of the MLP has 13 nodes corresponding to 13 MFCC parameters in a single frame, and the output layer contains one node corresponding with one of the 14 SPE features[34].

3.4.1.2 Multiclass MLPs for Intl. Phonetic Assoc. (IPA) classes

Several multiclass MLPs, each with 1000 hidden nodes and between 3 and 9 output nodes, were used to detect 44 phonetic attributes as defined by the International Phonetic Association. The inputs are 13 perceptual linear predictive (PLP) features and their 1st- and 2nd-order time derivatives within a 9-frame window, including 4 frames each of preceding and following context. We trained 8 MLPs separately, each representing one phonological class from the IPA chart (sonority, voicing, etc) with several possible values. The Voicing class, for example, has labels: Voiced, Voiceless and N/A. These labels correspond to the three output nodes for the Voicing MLP. The N/A label is used to form an exhaustive class set when necessary.

Details of the eight MLPs used to detect the IPA attributes are given in the last row of Table 2. Collectively, the eight MLPs have 44 output nodes. We use each of these outputs as an individual attribute detector in the ASAT framework.

3.4.1.3 HMM-based attribute detectors

Conventional hypothesis testing is based on the Neyman-Pearson lemma which uses the likelihood ratio to accept or reject a proposed hypothesis. A generalized likelihood ratio is computed when a test observation $\mathbf{O}$ is observed, and then compared against a decision threshold to decide which of two hypotheses is to be accepted. In order to conduct the test, one needs knowledge of two probabilistic models (for the null and alternative hypotheses), which are conventionally obtained through distribution estimation using pre-labeled data of sufficient amount. For the attribute detection problem, we model the null and alternative hypotheses with the well-known hidden

Markov model.

We model each of the 17 phonetic attributes listed in Table 2 (last column, second row) with a *pair* of HMMs. Each target phonetic attribute and an "anti-target", is modeled with a 3-state, 16-mixture HMM. A 2-class recognition is first performed on the speech signal, using just the target HMM, to obtain segments. Both HMMs are then Viterbi-aligned to each segment. For an observation $\mathbf{O}$, the detector score is computed as the log-likelihood ratio $LLR(\mathbf{O}) = \log \mathbf{L}(\mathbf{O}|\lambda_0) - \log \mathbf{L}(\mathbf{O}|\lambda_1)$ where $\log \mathbf{L}(\mathbf{O}|\lambda_0)$ and $\log \mathbf{L}(\mathbf{O}|\lambda_1)$ are acoustic likelihoods of the target and anti-target models, respectively [52, 81].

3.4.1.4 Phonetic boundary detection

Regions near phone boundaries and phonological feature boundaries may carry rich and important information for speech recognition. We attempted to extract boundary information and integrate this type of attribute into ASR systems as supportive information. Acoustic features (12th order PLP coefficients and their derivatives), and estimated probabilities of phones and phonological features were used as input features to our boundary detectors. For each of the 8 broad phonetic classes listed in Table 2 (last column, third row), a fully connected Multi-Layer Perceptron (MLP) with 4 output nodes was developed to detect transitions between the classes, resulting in 32 attributes for phonetic boundary detection. The 4 output nodes for each MLP classify a frame of speech as a Left Boundary (LB), Right Boundary (RB), Non-Left Boundary (NL) or Non-Right Boundary (NR) [89].

3.4.2 Detector Performance

A compelling advantage of the detection-based ASR paradigm and the use of bottom-up knowledge integration is that the standalone performance of low-level detectors of knowledge sources can be evaluated. In this section we briefiy evaluate the performance of detectors of knowledge sources in the context of detection-based ASR.

The Detector Error Trade-off (DET) curve, much like the ROC curve, plots the locus of a detector's accuracy over the complete range of threshold values to examine the trade-off between the number of false alarms and misses a detector will produce, and was used extensively in the development of speech attribute detectors in the ASAT project.

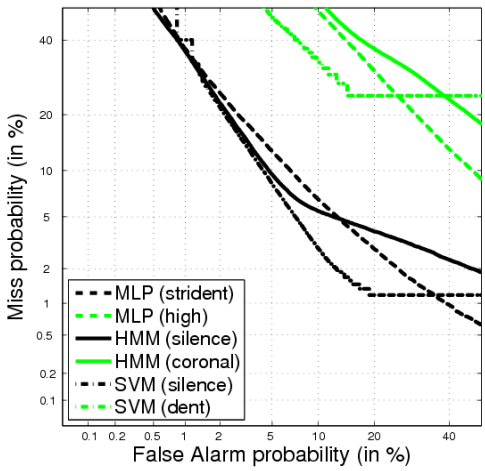


Figure 5: Selected Detector Error Trade-off (DET) curves for 2-class MLP, HMM and SVM detectors.

Selected plots of the Detector Error Trade-off (DET) curve are given in Figure 5. The plots represent the best and worst performing attributes for the HMM, 2-class MLP and SVM[1] detectors. HMMs and SVMs perform best for detecting *silence* while 2-class MLPs detect the *strident* attribute best. The best and worst equal error rates (EER), are given in Table 3. At the EER, the miss rate and false alarm rate are equal. All experiments are carried out on the TIMIT database.

Table 3: Minimum and maximum Equal Error Rate (EER).

Methods of detection	max EER	min EER
2-class MLP	0.250 (high)	0.081 (strident)
HMM	0.286 (coronal)	0.069 (silence)
SVM	0.417 (mid)	0.060 (silence)

[1]The worst performing SVM detector (mid) is not shown in Figure 5

3.4.3 Continuous Phone Recognition Results

Conditional Random Fields (CRFs), are discriminative models for sequences that attempt to model the posterior probability of a label sequence conditioned on a set of input observations. A CRF defines the conditional probability $P(\mathbf{Y}|\mathbf{X})$ as: $P(\mathbf{Y}|\mathbf{X}) = \exp \sim_t (\sum_i \lambda_i f_i(\mathbf{Y}, \mathbf{X}, t))/Z(\mathbf{X})$, where $\mathbf{Y}$ is a sequence of labels, $\mathbf{X}$ is a set of input observations, each function f is a feature function with an associated λ-weight, and the term $Z(\mathbf{X})$ is a normalizing term computed over all label sequences.

As in [61], CRFs were used in the ASAT project to perform continuous phone recognition on the TIMIT speech database using only the attribute detectors discussed in the previous section as inputs. Table 4 gives results of continuous phone recognition experiments performed using CRFs with several configurations of speech attribute detectors as inputs. The results in Table 4 are sectioned into 3 groups. Experiments in the first group involve using just one of the sets of attribute detectors in Table 2. Among these, the best performance, a phone accuracy of 68.96%, is achieved with multi-class MLPs.

Table 4: Continuous phone recognition experiments with conditional random fields on TIMIT.

Attribute Detectors	No. of Attrs.	Accuracy (%)
Multi-class MLP (MC-MLP)	44	68.96
HMM	13	46.14
SVM	17	42.83
2-Class MLP (2C-MLP)	14	46.51
MC-MLP, HMM	44+13	68.56
MC-MLP, SVM	44+17	69.29
MC-MLP, 2C-MLP	44+14	69.15
MC-MLP, HMM, 2C-MLP	44+13+14	68.54
HMM,Phonetic Feature Boundaries (PFB)	13+32	51.50
MC-MLP, PFB	44+32	69.02
MC-MLP, SVM, PFB	44+17+32	69.26
MC-MLP, HMM, PFB	44+13+32	70.47
MC-MLP, HMM, 2C-MLP, PFB	44+13+14+32	70.63

In the second and third groups in Table 4, two or more sets of attribute detectors are used as inputs to the CRF system. In the second group, all detectors are combined with multi-class MLPs, the best performing attribute detectors from the first group, and an accuracy of 69.29% is obtained in the best case. Finally, we incorporate phonetic boundary detectors in the last group in Table 3. The best phone accuracy result, 70.63%, is obtained when HMM-based detectors, multi-class MLPs, 2-class MLPs and phonetic feature boundaries are all incorporated, making use of 103 knowledge scores. The results of this first set of experiments are very encouraging since, in our detection-based framework, there is always room to incorporate more knowledge sources.

3.5 Discussion

In this chapter, we conducted experiments in detecting evidence of speech motor activity in speech audio signals. We used support vector machine (SVM) classifiers to detect attributes of the speech signal related to the articulators used, and the manner and place of articulation. Classifiers were trained for each speech attribute and tested in parallel on speech data to produce an ensemble of continuous-valued confidence scores. Confidence scores for each speech attribute were evaluated with receiver operating characteristics curves to illustrate the trade-offs between miss and false alarm errors.

This work was part of the Automatic Speech Attribute Transcription (ASAT) project; a recently proposed approach to automatic speech recognition based on detecting phonological attributes of speech. In addition to our SVM-based detectors, attribute detectors based on multi-layer perceptron artificial neural nets and hidden Markov models were used by our collaborators to detect phonological speech attributes as well. Confidence scores from these attribute detectors were combined in various configurations by our collaborators using conditional random fields. An

accuracy rate of 70.63% was obtained in the continuous phone recognition task using more than 100 detectors.

3.6 Conclusions

We have used support vector machine classifiers to detect encoded evidence of articulation gestures and other phonological attributes in speech audio signals. Comparable performance to the state-of-the-art was obtained in the continuous phone recognition speech recognition task, demonstrating that indirect evidence of speech motor activity can be used to decode intended speech content.

CHAPTER IV

CLASSIFICATION AND DETECTION OF NEURAL DATA FOR A NEURAL SPEECH PROSTHESIS

In this chapter we investigate decoding methods for an intracortical neural speech prosthesis. We use intracortical neural data collected in a related study from a human volunteer, hereafter referred to as "ER," living with "Locked-In Syndrome." The study involved the implantation of a microwire electrode in speech motor cortex of ER's brain and was conducted largely by colleagues at Neural Signals Inc., Duluth, GA and Boston University. Details of the subject, surgical implantation procedure and hardware for real time neural signal acquisition are given in Section 4.1. We discuss our approach to classifying neural data previously collected from ER while performing controlled tasks of imagined speech production. We use statistical methods to classify these data into a discrete set of vowel classes and to detect attempted speech activity.

4.1 Subject and Implant

A 26 year old male with tetraplegia (including loss of vocal or facial muscle control) as a result of Locked-in Syndrome volunteered for implantation with the Neurotrophic Electrode [42, 3] in 2004. Both the subject and his guardian provided informed consent for the intracortically implanted speech BCI study. The implantation procedure was approved by the Food and Drug Administration (IDE G960032), Neural Signals, Inc. Institutional Review Board and Gwinnett Medical Center Institutional Review Board. The purpose of the study was to investigate the neural mechanisms involved in

speech production and provide a communication mechanism through extracellular microelectrode recordings of speech-related activity in the motor cortex. A pre-operative functional magnetic resonance imaging (fMRI) study of imagined picture naming and word repetition was used to localize the optimal brain region of interest for implantation of the Neurotrophic Electrode. The results of this study indicated a location on the ventral precentral gyrus with the maximum activity during the attempted speech production tasks. Further details of the electrode and implantation procedure can be found elsewhere [3, 28].

Figure 6: Receiving antenna for wireless transmission of extracellular electric potentials.

Extracellular potentials were recorded with the Neurotrophic Electrode, wirelessly transmitted across the scalp and acquired with a Neuralynx Cheetah (Bozeman, MT) data acquisition system at 30303 Hz sampling rate, with 16-bit A/D resolution. The extracellular signals, on the order of 10–50μV, are bandpass filtered between 300 – 6000 Hz and amplified in hardware before being wirelessly transmitted across the scalp using an FM radio system. A picture of the receiving antenna placed over the scalp for wireless transmission of extracellular electric potentials is given in Figure 6.

A system diagram for the neural speech prosthesis used in these experiments is illustrated in Figure 7. After wireless FM transmission, amplification and signal processing operations, real-time spike-sorting using a manual cluster-cutting technique is applied to extracellular electric potentials acquired from the Neurotrophic electrode; this is described in more detail in Section 4.2. The result of the spike-sorting operation is an ensemble of neural spike trains, which comprises the primary input signal for neural decoding operations. Methods of neural decoding or classification are implemented in real time on a personal computer and subject ER is given audio and visual feedback based on the decoded result. Visual feedback is displayed on a large screen in front of the subject.

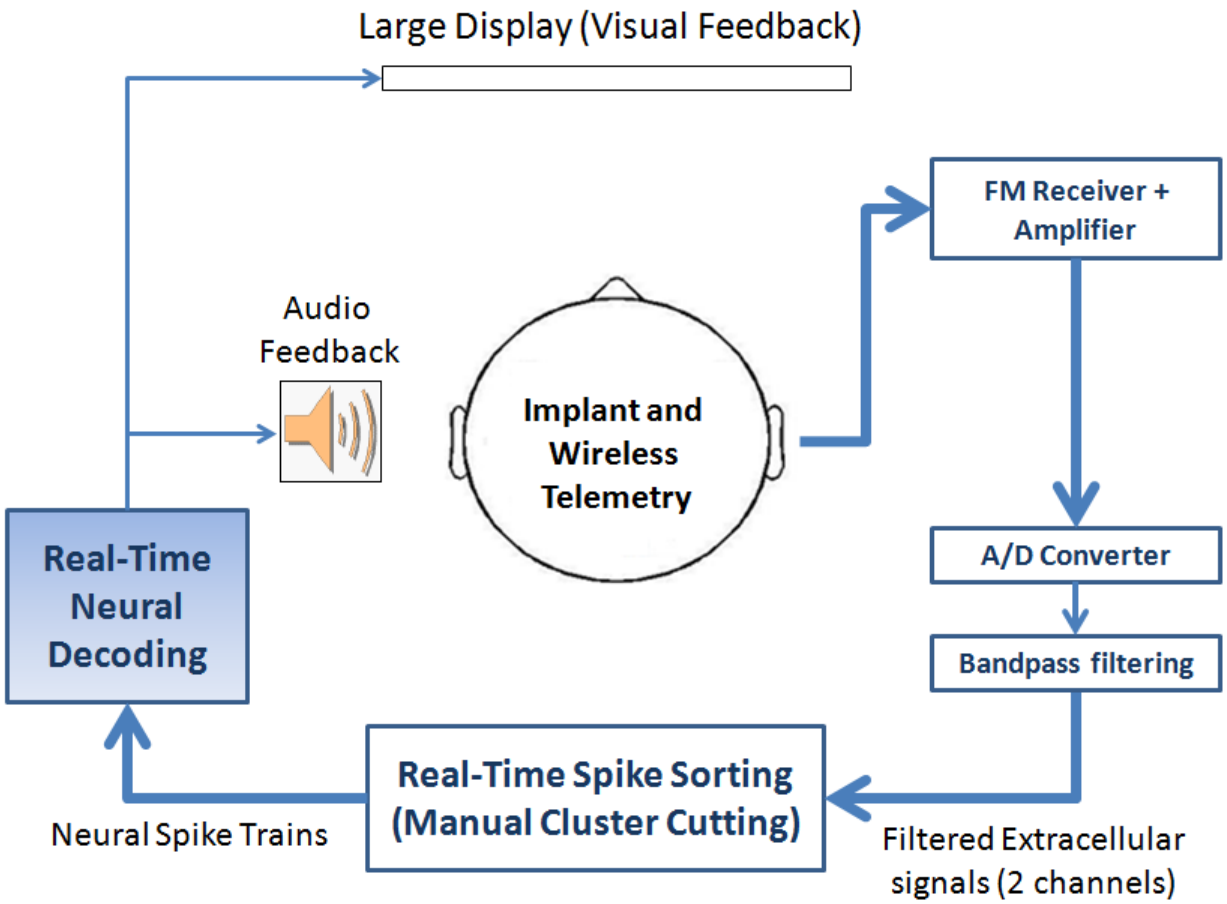


Figure 7: Neural Speech Prosthesis system diagram.

4.2 Manual Spike Sorting (Cluster Cutting)

Spike sorting for extracellular signals acquired by the electrode is done according to a manual sorting or cluster cutting technique for all experiments in this chapter. When possible, single units (i.e., clusters of neural data corresponding to a single neuron) were isolated using features of the spike waveform, and distinct multi-unit clusters were obtained when single unit detection was impossible. For the remainder of this discussion, all references to neuronal "units" or "clusters" encompass both single- and multi-units.

Spikes in the extracellular signal were detected as crossing above or below a threshold of $\pm 10\mu V$. For each spike, a 32-sample (1.05 ms) waveform is collected with the peak amplitude aligned to the 8^{th} sample. Neural units were defined by an expert via a convex-hull technique using spike peak and valley amplitude. In Chapter 6, a scatter plot of these features is given in the upper panel of Figure 33 for 3 putative neural units. Clusters are defined manually on a computer screen by drawing a polygon or other closed curve shape in the feature space. Putative neural units were successively split based on meaningful statistics of the firing rate associated with each cluster, particularly inter-spike interval histograms, and cross-correlations between putative firing times. Since individual units with uncorrelated firing rates were preferred, clusters were split until an increase in the cross-correlation between putative units was observed. Superimposed plots of 1 ms action potential waveforms for 20 single- and multi-unit clusters collected over a 12-second period are shown in Figure 8.

For all neural data used in this chapter, 56 neural units were identified on the 2-channel Neurotrophic Electrode; 29 units on channel 1, and 27 units on channel 2. Complete details concerning the data acquisition system can be found in [3, 28].

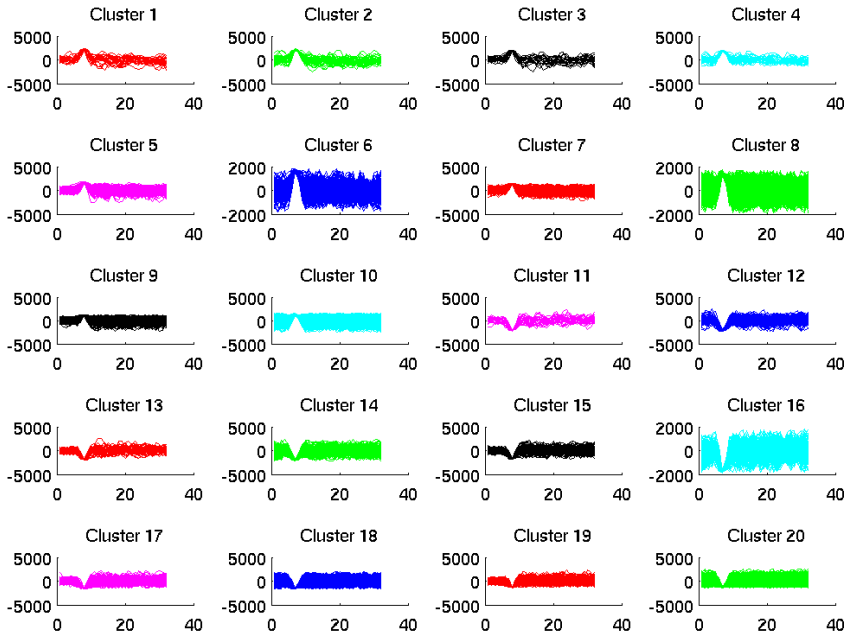


Figure 8: Superimposed action potential waveforms for 20 single- and multi-unit neuronal clusters on Channel 1.

4.3 Firing Rate Estimation

Given an extracellular trace in an intracortical BCI, the end result of applying a spike-sorting or cluster cutting procedure is an ensemble of "spike trains," one for each neuronal cluster. Many approaches to the analysis and decoding of neural spiking activity are based on modeling the ensemble of spike trains as a set of inhomogenous Poisson Processes, each completely specified by its time-varying firing rate parameter $r(t)$. In this sense, we completely model the information about the stimulus or activity of interest by the underlying neural firing rate with respect to time. As a result,methods of *estimating* the underlying firing rate, given an observed spike train, are incorporated in most intracortical BCI systems.

Though the basic idea is simple, the problem of firing rate estimation in BCIs has many important challenges, especially for neural prosthetic systems. Generally, firing

rate estimates should accurately characterize time-varying neural spiking activity, producing a measure that is higher when spikes fire more frequently and lower in sparse firing. In the context of decoding neural data, the firing rate measure should be continuous and smooth for suitability with many pattern classification methods. Furthermore, if the neural response to a stimulus or activity of interest is reflected in a neural spike train, it should also be reflected in the firing rate measure. This can be tested more easily in cases when a stimulus is applied directly to a population of neurons, as with electrical stimulation. For neural prosthetic systems, however, the activity of interest is often indirect (as with motor movement) or even imagined (as with a prosthesis for speech).

In the remainder of this section, we describe the firing rate estimation methods we use in the analysis and classification of neural data in the context of a neural speech prosthesis. We discuss the simple histogram or binning method based on counting neural firings in short time windows, the kernal smoothing method in which spike trains are convolved with a smooth kernel function, and the adaptive exponential method, a recently proposed non-stationary parametric firing rate method. Generally, we will refer to the true firing rate as $r(t)$, and its computed estimate as $x(t)$.

4.3.1 Histogram method (Binning)

The firing rate $r(t)$ for a given cluster is an instantaneous measure of the intensity of its firing activity. A simple and effective way to compute firing rates is to use counts or histograms of firing events in bins of equal lengths. For a single spike train, we can define the firing rate *estimate* at time t as

$$x(t) = \frac{N(t) - N(t-\tau)}{\tau}, \tag{53}$$

where $N(t)$ is the total count of neural firings up to time t. The rate estimate is then the count of firing events in a τ-length time window ending at time t, normalized

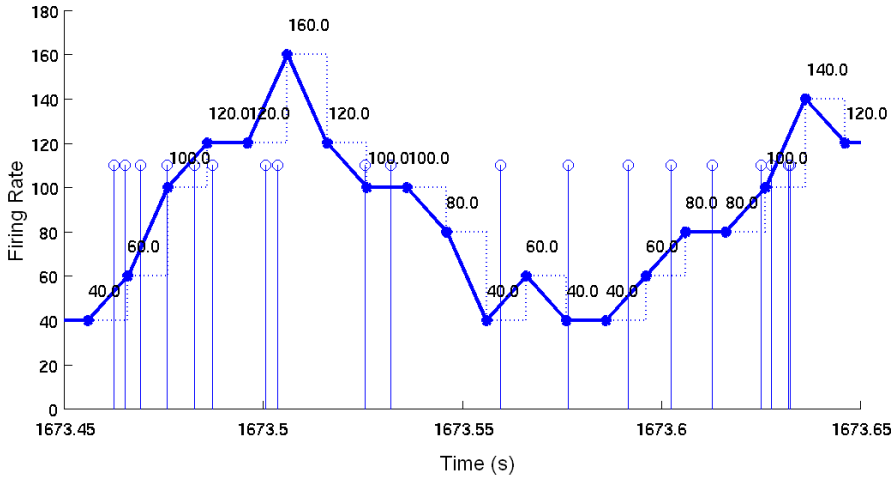


Figure 9: Time-varying firing rate estimate for a neural spike train (depicted with stems) using the histogram method. Bin length is 50 ms and rate estimates are spaced 10 ms apart.

by the window length. The histogram or binning method for firing rate estimation has two simple parameters: the window length τ and the update interval between estimates Δt. A plot of 0.2 second of a neural spike train along with corresponding firing rate estimate based on firing event counts is given in Figure 9. The firing rate estimate depicted in Figure 9 is based on counts of firing events in overlapping $\tau = 50$ms windows updated in $\Delta t = 10$ms time frames, and has units of *events per second*. The actual value of the instantaneous firing rate estimate is printed above each point in the plot in Figure 9. This is to illustrate that, while this method is simple and effective, the resulting signal is not smooth and its values are coarsely quantized.

4.3.2 Kernel Smoothing

While the histogram method is intuitive and computationally efficient, its effectiveness is limited by several shortcomings. Perhaps most significant is that the firing rate measure it produces is not continuous and smooth and, as a result, not well suited for many statistical classifiers, such as Gaussian mixture models. A common approach

for producing smooth, continuous-valued firing rate estimates is to convolve the spike train with a smooth kernel function [16]. Modeling the i^{th} spike train $\mathbf{t}_i = \{t_{i,j}\}_{j=1}^{N_i}$ as an impulse train $y(t)$ such that

$$y(t) = \sum_{j=1}^{N_i} \delta(t - t_{i,j}), \tag{54}$$

the firing rate $x(t)$ is given as follows

$$x(t) = \int y(t - \tau)k(\tau)d\tau, \tag{55}$$

where $k(t)$ is the smooth kernel function. In all experiments in this chapter, we use a Gaussian kernel function of the form

$$k(t; \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{t^2}{2\sigma^2}\right), \tag{56}$$

where σ, which is called the "radius" or "bandwidth" parameter, determines the shape (particularly the effective width) of the kernel function. The firing rate $y(t)$ is the weighted average of the time difference between t and the nearest spike occurrence times.

4.3.3 Adaptive Exponential Method

A novel method of firing rate estimation, which we will refer to as the adaptive exponential method, was introduced in [10]. The firing rate is updated at each firing event so that, given a spike train $\mathbf{t} = \{t_k\}_{k=1}^{N}$, we define $y_k = y(t_k)$, the firing rate at time t_k, to be

$$y_k = \log\left(1 + e^{y_{k-1} - \gamma\delta_k/\tau_{k-1}}\right) \tag{57}$$

$$\tau_k = \tau_{k-1} + (1 - e^{\epsilon\delta_k}) \cdot (\delta_k - \tau_{k-1}), \tag{58}$$

where τ_k, an intermediate variable, is the time-varying exponential rate parameter and $\delta_k = \tau_k - \tau_{k-1}$. The parameter ϵ is the learning rate for τ_k and γ is a scaling

factor. Generally, the firing rate at any time t is given by

$$y(t) = \log\left(1 + e^{y_{k-1} - \gamma(t-\tau_{k-1})/\tau_{k-1}}\right). \tag{59}$$

The adaptive exponential method, with its time-varying rate parameter τ_k, is a non-stationary firing rate estimation method. The rate parameter τ_k adapts to changes in the underlying spike rate based on δ_k, the elapsed time since the last firing event. τ_k adapts slowly when the underlying firing rate is low and quickly when firing rate is high. As a result, the firing rate estimate is normalized across wide variations in the underlying firing rate, which is desirable for using neural data recorded on different dates or under significantly varying conditions. However, the output quantity $y(t)$ for the adaptive exponential method is not a meaningful measure of the firing rate.

4.4 Experimental Results

4.4.1 Data

We evaluate our probabilistic approach to neural decoding using data collected in a previous study by Brumberg, Kennedy and Guenther conducted with subject ER [10, 28]. The work presented in [10] constitutes the first known intracortical neural prosthesis for speech. In these experiments, the subject ER, implanted with the Neurotrophic Electrode, performed imagined speech production tasks and was able to successfully operate a speech synthesizer using only neural control. In each single trial, one or more synthetic vowel sounds are first played for the subject in a "Listen" phase. After a short pause, a bell is played for the subject and the "Speak" phase begins in which he attempts to repeat the same vowel sounds using the neural speech prosthesis. The vowel sequences in each trial consisted of transitions between two vowels. This took the form of either a "2-state" transition V1→V2, or a "3-state" transition V1→V2→V1. These 2 recording paradigms are illustrated in Figure 10.

We use neural data from these experiments, recorded on 3 dates in 2008 (1/11/2008,

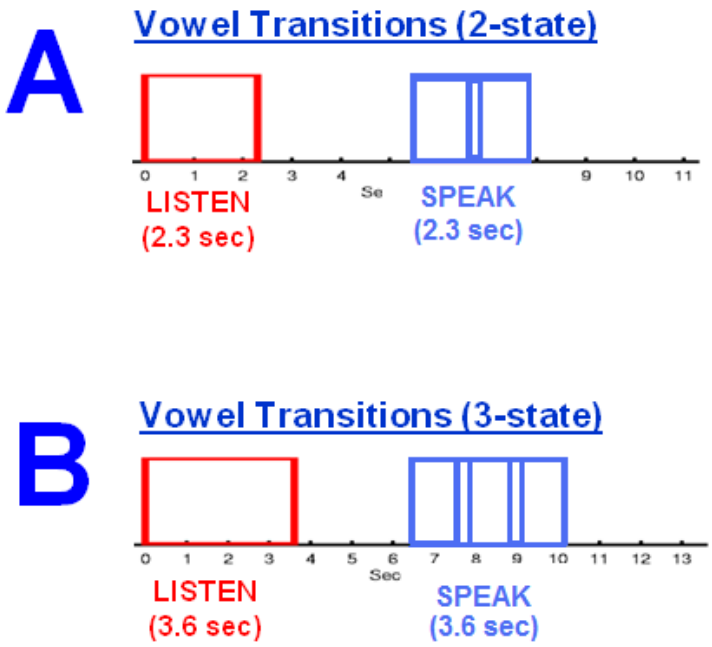


Figure 10: Recording paradigms for continuous vowel data.

5/19/2008, and 10/1/2008). A breakdown of the number of trials in each dataset and the recording paradigms used is given in Table 5. Available data from the recordings include continuous extracellular signals, and ensemble firing times for the neural units as determined by the manual spike-sorting and subsequent spike classification.

Table 5: Recording dates and descriptions for vowel decoding data.

Date	No. Trials	Trial	Size (s)	Vowels	Chance Level	Error Margin
1/11/2008	45 Trials	B	612 s	/aa/,/ae/,/ah/,/uw/	58.3%	0.49%
5/19/2008	40 Trials	A	372 s	/aa/,/ah/,/iy/,/uw/	44.1%	0.54%
10/1/2008	48 Trials	A	447 s	/aa/,/ah/,/iy/,/uw/	44.1%	0.47%

4.4.2 Methods

After applying the single- and multi-unit clusters defined by the manual cluster cutting procedure in Section 4.2 to neural data collected from ER, the result is an

ensemble of 56 neural spike trains for each trial in each data recording session. A plot of the average firing rate for each neural unit is given in Figure 11 for the dataset recorded on 10/1/2008, computed as the total count of neural firings divided by the full length of the data set. Figure 11 illustrates the differences in firing rates among the neuronal clusters; some fire very frequently (e.g., Channel 2, Cluster 17), while others rarely fire (e.g., Channel 1, Cluster 1).

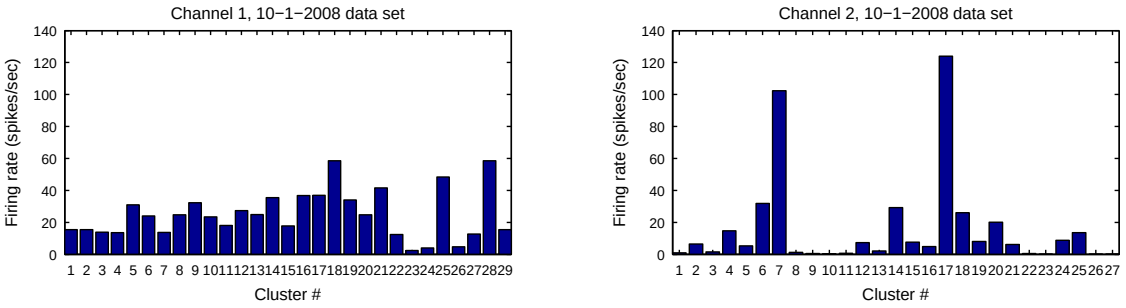


Figure 11: Average firing rates for 56 neural units (29 and 27 units on channels 1 and 2, respectively.

Our approach to analyzing and classifying neural data is to apply statistical pattern recognition methods to time-varying estimates of the firing rate during imagined speech production offline. We define the variable $\mathbf{T} = (\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_D)$ to be the ensemble of neural spike trains, where D is the number of neuronal clusters ($D = 56$, in this case) and each $\mathbf{t}_i = \{t_{i,j}\}_{j=1}^{N_i}$ is a point process consisting of the N_i firing occurrence times for the i^{th} neuronal cluster. The result of applying firing rate estimation to each spike train in $\mathbf{T}$ is an $N \times D$ matrix $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_N\}$ consisting of N, D-length continuous- or discrete-valued instantaneous firing rate estimates, equally spaced in time. Finally, we classify multivariate firing rate estimates in $\mathbf{X}$ using a Gaussian mixture model classifier for two tasks: frame classification for short duration frames of neural data and speech activity detection.

4.4.3 Instantaneous Neural Firing Rates

We apply the methods in Section 4.3 to offline neural spike train ensemble data to obtain instantaneous neural firing rate estimates for each spike train. Raster plots of neural firing times, along with several plots of instantaneous firing rate estimates, are given in Figure 12 for two neural units, with very different firing characteristics, taken from the Oct-01-2008 data set over the same 4.0 second period. The neural units shown in the figure are Channel 1, Cluster 1, and Channel 2, Cluster 17 in the lower panel, and have average firing rates of 15.46 and 123.95 spikes/s, respectively, as shown in Figure 11.

Firing rate estimate plots in Figure 12 include the adaptive exponential (AE) method with parameters $\gamma = 0.25$, $\epsilon = 0.50$ and $\tau_0 = 2.0$ and the kernel smoothing method using a Gaussian kernel with bandwidth parameters $\sigma = 50$ ms (KS50), $\sigma = 100$ ms (KS100), $\sigma = 150$ ms (KS150), and $\sigma = 250$ ms (KS250); note that each of these plots has been scaled to a maximum value of 1 for comparison in the figure. For both the low and high firing rate spike trains in Figure 12, the firing rate is non-stationary. This observation is clearly reflected for the low firing rate cluster in the upper panel of the figure as large variations in all firing rate methods are observed. For the high firing rate neuron, high bandwidth kernel smoothing methods KS150 and KS250 and the adaptive exponential method remain relatively constant while rapid fluctuations are observed in KS50 and KS100.

Histograms of firing rate estimates for the high and low firing rate neurons are given in the upper left and upper right panels of Figure 13, respectively. For all of the kernel smoothing methods, a mode in the histogram near the average firing rate is distinguishable by inspection. The adaptive exponential method, however, does not reflect a meaningful estimate of the true firing rate, due to its self-normalizing property. This is more clearly illustrated in the lower left and right panels of Figure 13 where histograms of the log of the firing rate are plotted for each method.

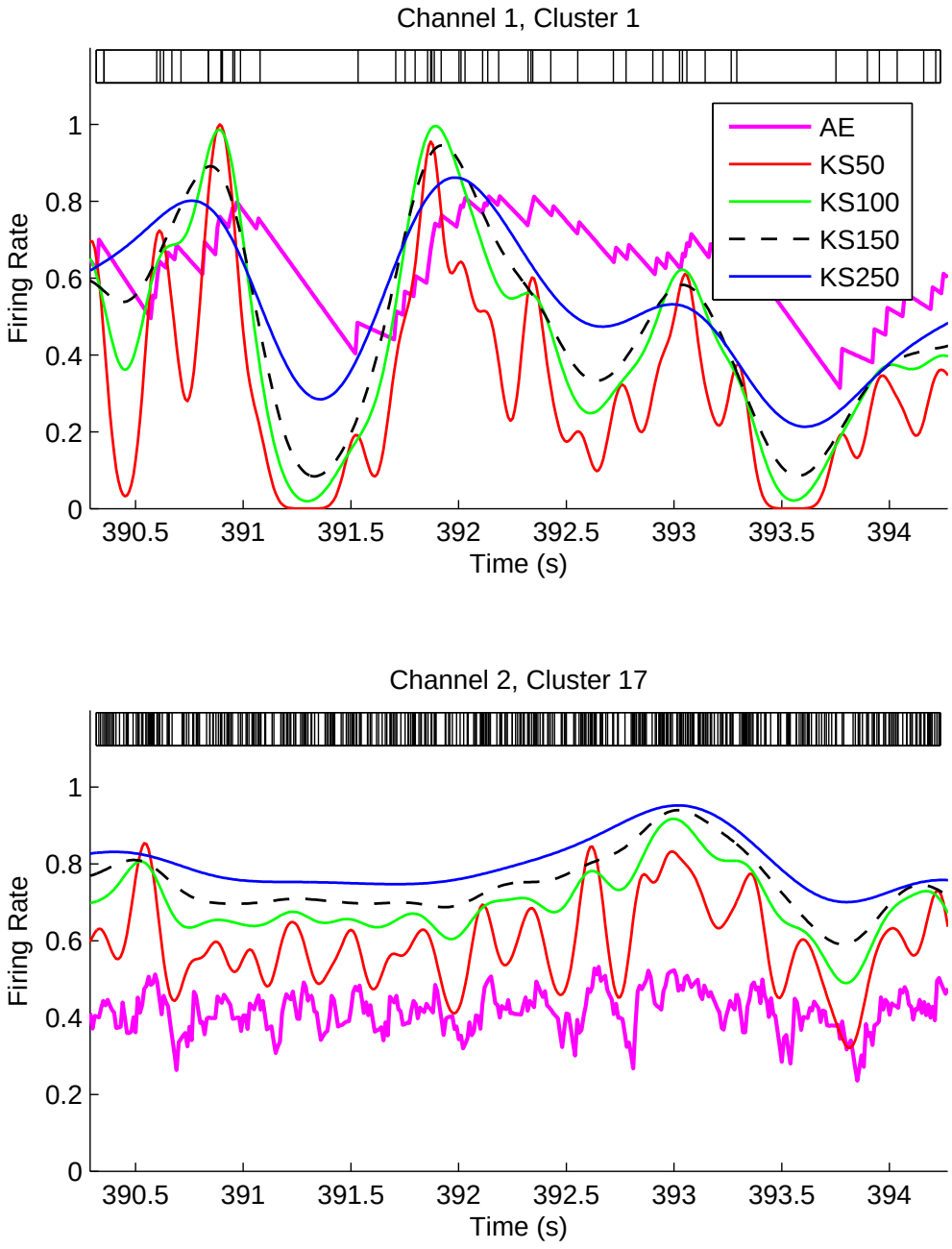


Figure 12: Spike raster plot (inset and above in each panel) and instantaneous firing rate estimates for a fast firing (upper panel) and slow firing (lower panel) neural unit for a 4.0 second period. Firing rate methods shown include the adaptive exponential (AE) and kernel smoothing methods for $\sigma = 50$ ms (KS50), $\sigma = 100$ ms (KS100), $\sigma = 150$ ms (KS150) and $\sigma = 250$ ms. Each plot is normalized to a maximum value of 1.0.

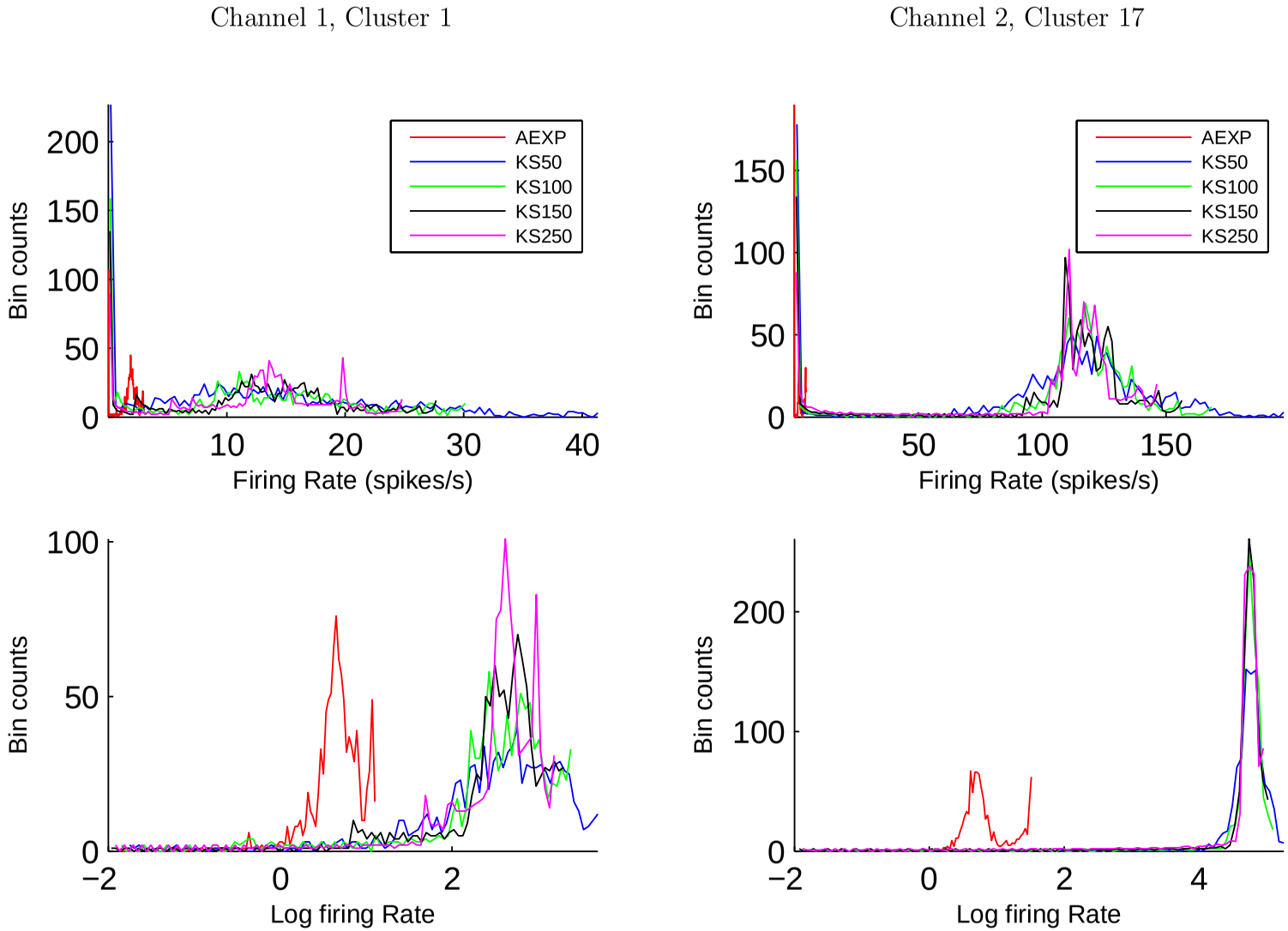


Figure 13: Histograms of instantaneous firing rate estimates (upper panels) and log firing rates (lower panels) for the slow (left) and fast (right) firing neural units in Figure 12. Adaptive exponential (AEXP) firing rate estimates for fast and slow neurons are distributed over a similar range due to its normalizing property. See Figure 12 for plot legend abbreviations.

4.4.4 Frame Classification

We apply statistical pattern recognition methods to instantaneous neural firing rate estimates in the context of a neural prosthesis for speech. We use the binning, kernel smoothing and adaptive exponential methods to estimate neural firing rates every $\Delta t = 10$ms for $D = 56$ neuronal clusters. We classify each frame of neural data into a discrete set of classes including 5 vowels (/aa/, /ae/, /ah/, /iy/, /uw/) and a class for silence or non-speech. A list of vowel classes for each data set, along with the number of individual trials, is given in Table 5.

On each recording date, the set of classes is 4 vowels + "non-speech"; this would give an 80% error rate (i.e., a 20% accuracy rate) for guessing by chance with all 5 outcomes considered equally likely. However, after removing "Listen" segments, data content for Type A and B trials are 73.56% and 61.7% "non-speech," respectively, as indicated in Figure 10 and Table 5. We can estimate a more realistic (and, consequently, more difficult) chance level to achieve by assuming a chance process that guesses according to the known a priori probability of each class. The probability p of guessing correctly is then $p = \sum_k P(\text{true class} = k, \text{guess} = k) = \sum_k P(\text{true class} = k)^2$, where $P(\text{true class} = k)$ is the a priori probability of class k. For trial Type A, the chance error rate is $1 - [0.7356^2 + 4 \cdot 0.0661^2] = 0.4414$. Similarly, for Type B trials, the chance error rate is 0.5827; this is listed in Table 5 for the 3 recording dates.

In all experiments, we use Gaussian mixture model classifiers for multivariate firing rates, using the expectation-maximization algorithm for parameter estimation. Firing rate estimates for each neuronal cluster were first normalized to zero mean and unit variance before training and testing. We use a 5-fold cross-validation to evaluate performance. For each recording date, trials are randomly partitioned into 5 mutually exclusive sets; for each partition, we set aside 4/5 of the trials for a training set and the held-out fifth for a testing set.

Average classification error rate across all cross-validation folds is plotted in Figure 14 for training and testing data for 3 recording dates using the binning or histogram method. We vary the window length τ from 10 ms to 100 ms. We use a 16-mixture GMM classifier in all experiments; for GMM topologies with fewer mixtures, the coarsely quantized firing rate measures led to numerical instabilities in training. The error rate for both the training and testing sets is lowest for the longest window length $\tau = 100$ms across all data sets. The error rates at $\tau = 100$ms for the test set are 0.4154, 0.2962 and 0.3587 for the Jan-11-2008, May-19-2008 and Oct-01-2008 data sets, respectively, all performing significantly better than chance levels shown in Table 5.

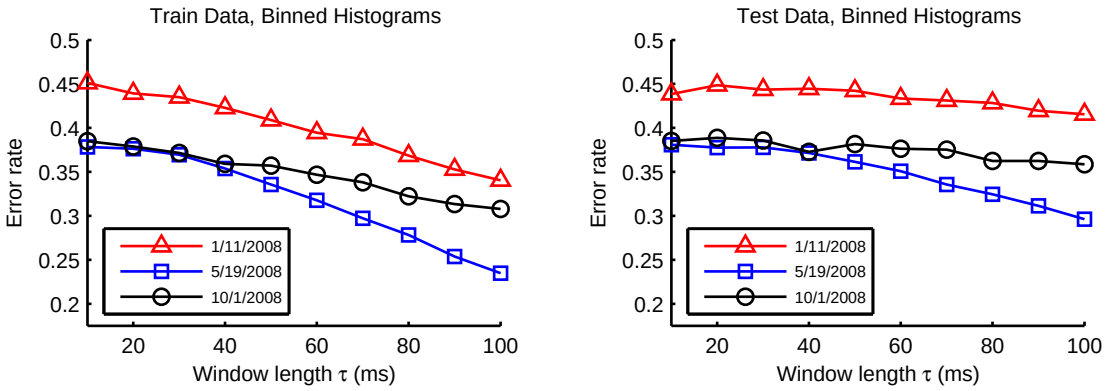


Figure 14: Vowel classification error rate for 10 ms frames of neural data using the histogram or binning firing rate method versus window length τ on 3 recording dates. Gaussian mixture models with $K = 16$ mixtures were used for classification. Chance levels for each recording date are given in Table 5.

Cross-validation error rates for the adaptive exponential firing rate estimation method and the kernel smoothing method with radius parameter $\sigma = 50$, 100, 150 and 250 ms are plotted in Figure 15 for 3 recording dates. Error rates are plotted for Gaussian mixture model classifiers with $K = 2$, 4, 8 and 16 mixture components. For the kernel smoothing method, the error rate generally decreases as the bandwidth σ

increases and, for the training data, as the number of Gaussian mixtures increases. For the testing set, performance for the kernel smoothing method is generally not affected by the number of Gaussian mixtures. The best performance for the adaptive exponential method for all recording dates is obtained with 4 Gaussian mixtures. The lowest test set error rates across all firing rate methods and classifier topologies for the 3 recording dates are 0.4753 (KS250, 2 mixtures), 0.3246 (KS250, 16 mixtures), and 0.3591 (adaptive exponential, 4-mixtures) for Jan-11-2008, May-19-2008 and Oct-01-2008, respectively. The lower left panel of Figure 15, shows that, of the 20 configurations of firing rate methods and classifier topologies, just 3 configurations perform better than chance on the testing data set. For the May-19-2008 and Oct-01-2008 datasets, 6 and 14 configurations, respectively, perform better than chance. As these 3 datasets are listed in chronological order and were recorded months apart, this may indicate that classifier performance improves with long term implantation of the device.

4.4.5 Speech Activity Detection

The task of speech activity detection, is common in automatic speech recognition systems intended for high noise environments. We define speech activity detection in the context of a neural speech prosthesis as reliably detecting when the subject is attempting to speak and rejecting all other "non-speech" activity in the neural data. Speech activity detection can serve as an important proof of concept for a discrete-state neural speech prosthesis as well as a redress for the so-called "Midas Touch" problem in brain-computer interfaces, where the user or subject is unable to turn the device "off" when desired [35].

As with the frame classification task, we model imagined speech activity using Gaussian mixture models trained on instantaneous neural firing rate estimates as

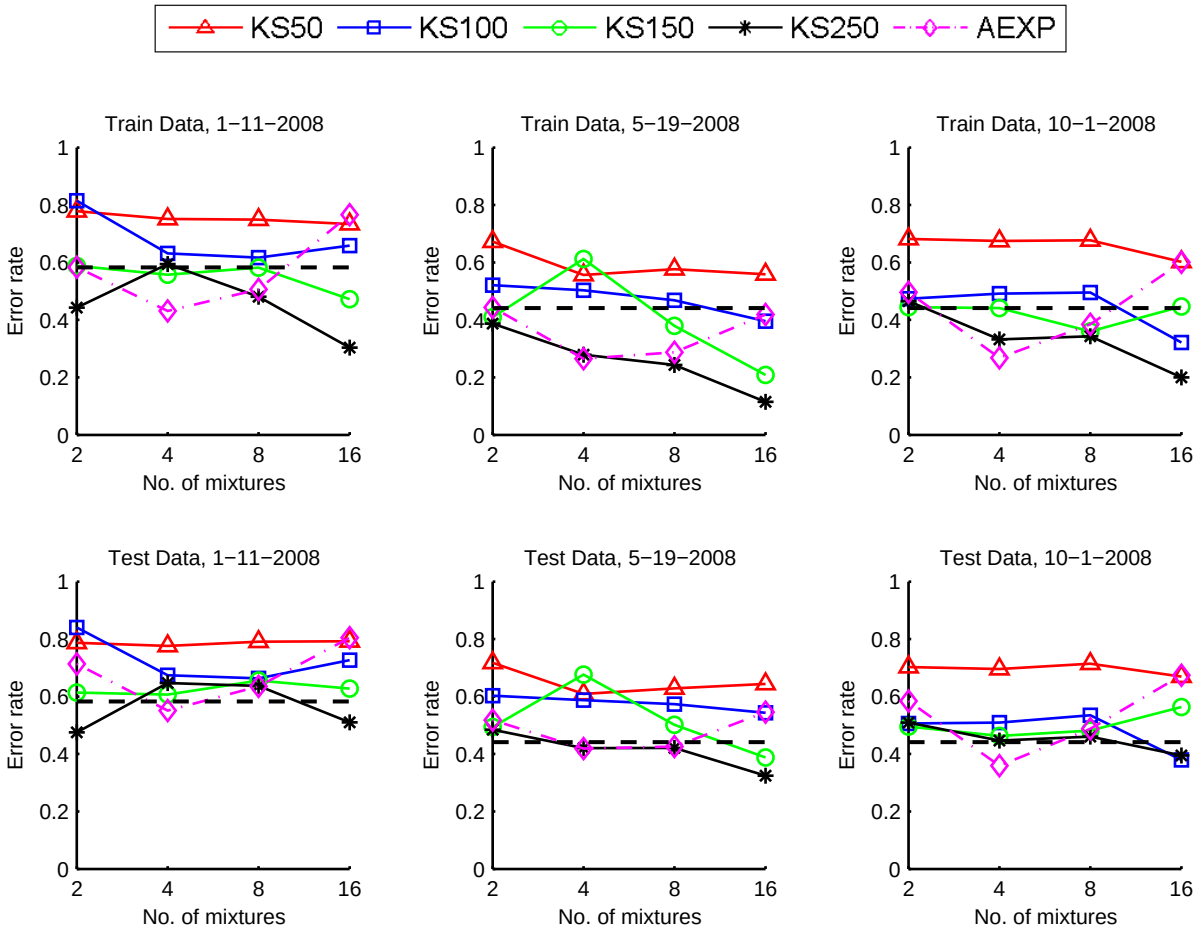


Figure 15: Vowel classification error rate versus the number of GMM mixtures using using adaptive exponential and kernel smoothing firing rate methods. Chance error levels for each recording date are indicated with black, horizontal, dashed lines. See Figure 12 for plot legend abbreviations.

described in Section 4.3. Illustrations of trial experiment paradigms are given in Figure 10. For the speech activity detection task, we define the class w_1 for portions of the trial where the subject ER is attempting to speak; other data belong to the w_0 or "nonspeech" class. Given a Gaussian mixture model with K mixture components, the likelihood is defined as $p(\mathbf{x}) = \sum_k c_k \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, where $\mathbf{x}$ is a vector of instantaneous firing rate estimates. We define the continuous-valued, 1-dimensional discriminant function $g_1(\mathbf{x})$ as the posterior probability of imagined speech production such that $g_1(\mathbf{x}) = p(\mathbf{x}|w_1)/(p(\mathbf{x}|w_0) + p(\mathbf{x}|w_1))$.

We use the adaptive exponential and kernel smoothing firing rate methods and

train Gaussian mixture models with K=2, 4, 8, and 16 mixture components to detect intended speech activity for the 3 recording dates. Using the discriminant $g_1(\mathbf{x})$ as a confidence score for speech activity, we evaluate detection performance based on the trade-off between Type I or "miss" errors and Type II or "false alarm" errors incurred as the decision threshold is varied. We report the equal error rate (EER) and minimum detection cost function (DCF) to evaluate performance. The detection cost function $C_{DET}(\theta)$ at a given threshold θ and DCF, the minimum overall cost function, are defined as

$$C_{DET}(\theta) = C_{miss} \cdot P_{Miss}(\theta) \cdot P_{target} + C_{FA} \cdot P_{FA}(\theta) \cdot P_{anti} \tag{60}$$

$$DCF = \min_{\theta} C_{DET}(\theta), \tag{61}$$

where $P_{Miss}(\theta)$ and $P_{FA}(\theta)$ are miss and false alarm probabilities at each threshold θ. P_{target} and P_{anti} are the a priori probabilities of "speech" and "non-speech," and C_{miss} and C_{FA} are the Type I and Type II error costs, respectively. We use $C_{miss} = C_{FA} = 0.5$ in all experiments.

EER and DCF values for training and testing data are are listed in Table 6 for 3 recording dates for all firing rate methods and GMM topologies. Values listed in the table are averages across 5 cross-validation folds. Speech detection performance on training data generally improves for the kernel smoothing method as the bandwidth σ is increased and as the number of Gaussian mixtures is increased for both the kernel smoothing and adaptive exponential methods. When evaluated on testing data, however, neither of these performance trends hold; very little change in performance is seen across firing rate methods and GMM topologies. The average minimum DCF values across all firing rate methods and GMM topologies for the testing data are 0.2980, 0.2533, and 0.2539 for Jan-11-08, May-19-08 and Oct-01-08 recording dates, respectively, with standard deviations of just 0.0028, 0.0035 and 0.0013.

Finally, receiver operating characteristic (ROC) curves are given in Figure 16 for

Table 6: Equal error rate (EER) and detection cost function (DCF) for the speech activity detection task on 3 recording dates GMM posterior probability as detection score for adaptive exponential and kernel smoothing firing rate methods. "Mix" is the number of GMM mixtures.

		Training set						Testing set					
Firing Rate Method	Mix	Jan-11-2008		May-19-2008		Oct-01-2008		Jan-11-2008		May-19-2008		Oct-01-2008	
		EER	DCF	EER	DCF	EER	DCF	EER	DCF	EER	DCF	EER	DCF
AEXP	2	0.3239	0.2665	0.2695	0.2424	0.3042	0.2502	0.3822	0.3000	0.3077	0.2508	0.3158	0.2541
	4	0.2717	0.2366	0.2440	0.2093	0.2636	0.2310	0.3643	0.2976	0.3147	0.2456	0.3061	0.2526
	8	0.2722	0.2184	0.1692	0.1550	0.2579	0.1903	0.3787	0.2971	0.2975	0.2497	0.3503	0.2518
	16	0.4413	0.2837	0.3314	0.1798	0.2542	0.1769	0.4726	0.2971	0.4261	0.2553	0.3415	0.2521
KS50	2	0.3462	0.2915	0.3156	0.2501	0.3600	0.2553	0.3801	0.2999	0.3394	0.2522	0.3605	0.2553
	4	0.4309	0.3001	0.3152	0.2550	0.3501	0.2535	0.4335	0.2999	0.3294	0.2544	0.3523	0.2546
	8	0.4250	0.2996	0.3189	0.2539	0.3464	0.2497	0.4269	0.3001	0.3411	0.2545	0.3543	0.2535
	16	0.3845	0.2999	0.2961	0.2476	0.3441	0.2465	0.3941	0.3000	0.3304	0.2555	0.3520	0.2547
KS100	2	0.2901	0.2427	0.3172	0.2555	0.3497	0.2553	0.3855	0.2973	0.3291	0.2544	0.3487	0.2553
	4	0.3939	0.2998	0.2933	0.2515	0.3359	0.2515	0.4141	0.2999	0.3168	0.2555	0.3506	0.2546
	8	0.3981	0.2927	0.2744	0.2385	0.3426	0.2487	0.4160	0.2999	0.3210	0.2549	0.3563	0.2552
	16	0.2566	0.2252	0.2596	0.2314	0.3345	0.2439	0.3733	0.2998	0.3219	0.2539	0.3596	0.2547
KS150	2	0.3728	0.2962	0.3128	0.2555	0.3425	0.2553	0.3895	0.2995	0.3287	0.2547	0.3437	0.2553
	4	0.3593	0.2916	0.2826	0.2455	0.3395	0.2462	0.3912	0.2999	0.3222	0.2555	0.3620	0.2543
	8	0.2487	0.2256	0.2464	0.2169	0.3276	0.2389	0.3605	0.2965	0.3377	0.2555	0.3473	0.2529
	16	0.1868	0.1609	0.2209	0.1961	0.2962	0.2292	0.3720	0.2947	0.3052	0.2534	0.3422	0.2547
KS250	2	0.3549	0.2915	0.3084	0.2511	0.3302	0.2553	0.3687	0.2960	0.3438	0.2555	0.3356	0.2553
	4	0.3560	0.2941	0.2731	0.2339	0.2978	0.2439	0.3883	0.2994	0.3455	0.2555	0.3304	0.2517
	8	0.3185	0.2705	0.2289	0.2129	0.2728	0.2238	0.3722	0.2888	0.3434	0.2555	0.3340	0.2523
	16	0.2998	0.2570	0.2007	0.1762	0.2612	0.2199	0.3757	0.2963	0.3120	0.2430	0.3338	0.2537

selected GMM topologies and firing rate methods. For each firing rate method, the ROC curve for the best performing GMM topology, in terms of equal error rate, is plotted in Figure 16. The curves illustrate that, even for the best performing configurations, which are represented in the figure, the performance of speech activity detection on unseen testing data is not significantly better than chance for the 3 recording dates.

4.5 Discussion

We conducted a proof-of-concept investigation for a neural speech prosthesis using Gaussian mixture models for classification and detection on extracellular signals extracted from speech motor cortex of a human subject. The data used in these analyses were collected in the course of a closely related study in which the subject participated in imagined vowel production. Using several firing rate estimation methods we classify short frames of neural data into vowel classes and detect attempted speech activity in the data as well.

Our choice of firing rate estimation methods was motivated by simplicity, computational efficiency and, in the case of the adaptive exponential method, by demonstrated success in a parallel study. Our motivation was simply to find the method best suited to classification and decoding tasks. For this reason we implemented several firing rate methods and judge success largely by the performance of the classification and detection tasks.

Among the 3 types of firing rate estimators we applied to the frame classification task, the lowest generalization (i.e., test set) error rate is achieved with the simple histogram or binning method, despite its stated disadvantages. With a 16-mixture GMM classifier, we obtain significant improvements over the chance level for the May-19-08 and Oct-01-08 data sets. We generally find that the error rate decreases as the

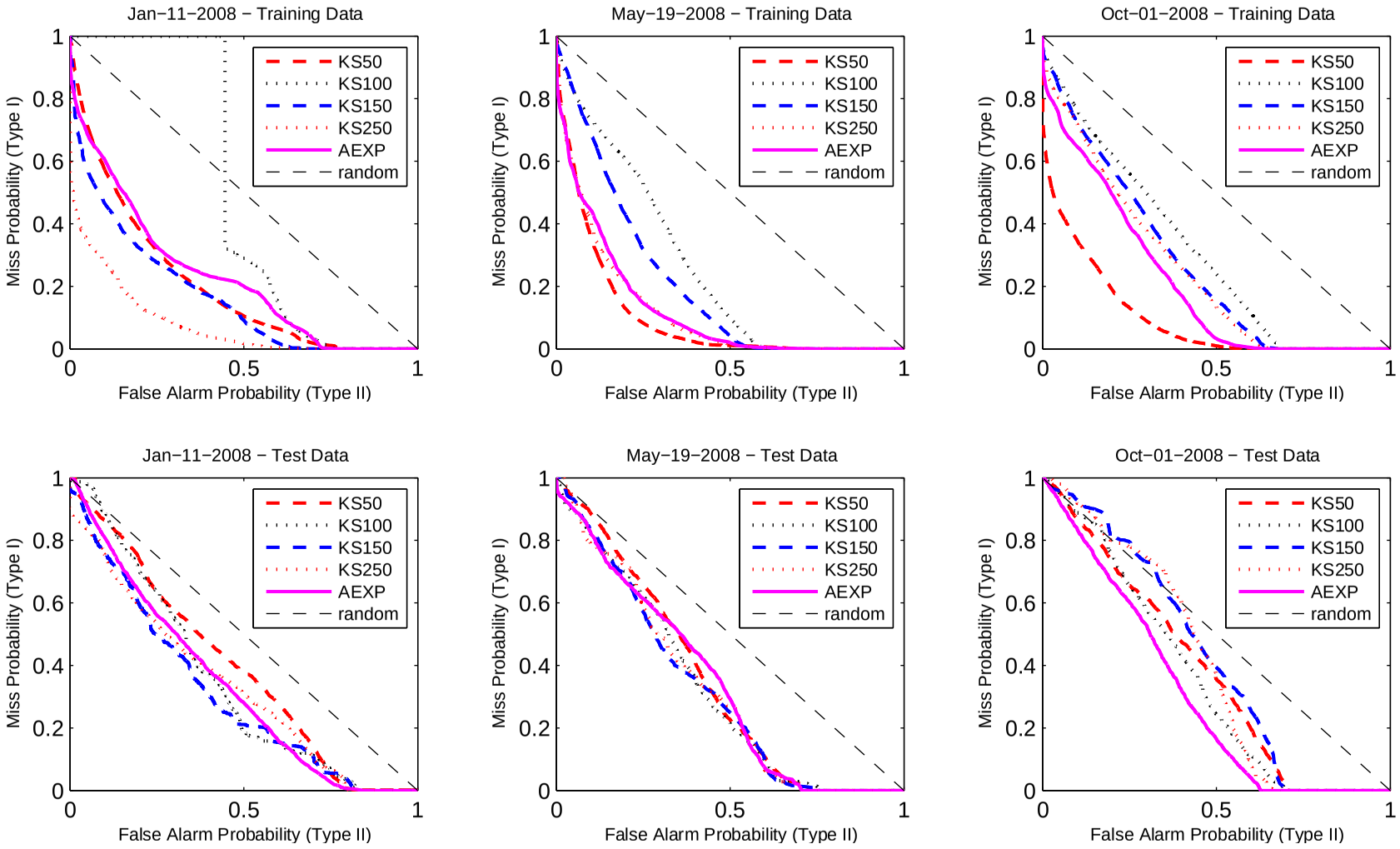


Figure 16: Selected receiver operating characteristic (ROC) curves for speech activity detection. ROC curves plotted for each firing rate method and dataset were selected as having the lowest EER in Table 6.

window length τ is increased for this firing rate method. However, the coarsely quantized firing rates led to numerical instabilities for GMM topologies with fewer than 16 mixtures. The kernel smoothing (KS) and adaptive exponential methods were applied to address this issue. The bandwidth parameter σ for the kernel smoothing method controls the effective analysis length of the kernel. Similar to τ for the histogram method, we generally find better performance with higher σ. One important limitation to using the KS method with a Gaussian kernel is that it is non-causal and would require a significant delay in a real-time neural speech prosthesis.

The adaptive exponential method is smooth and was demonstrated effective in the neural speech prosthesis study from which these data were collected. The non-stationary adaptive exponential method was designed to normalize fast and slow firing rates to a well-defined range and also to produce Gaussian distributed firing rate estimates well suited for use with a Kalman filter. While the lowest frame classification error rates obtained for the adaptive exponential method are comparable to the best performance for the other firing rate methods, performance is consistently worse for the higher order GMM classifier topologies; this result may be due to its normally distributed firing rate estimates.

It should be noted that the frame classification error rate as a performance metric does not immediately signify the performance of a neural speech prosthesis system. Much like an automatic speech recognition system, a fully functional neural speech prosthesis would need to decode neural data into a coherent, time-aligned output. However, frame classification performance is useful for comparatively evaluating classification methods, classifier topologies or firing rate estimation methods.

Unlike frame classification, the speech activity detection task and its metrics, the receiver operating characteristics (ROC) curve and corresponding DCF and EER measures, do signify meaningful performance measures for a neural speech prosthesis

system. Given the high-noise nature of cortically implanted electrodes and electronics, it is likely that isolating speech activity will be an important component of such a system. ROC plots on testing data, however, show that the generalization performance of speech activity detection is quite poor across firing rate methods and classifier topologies. The start and stop times for speech in each trial are a possible source of variation in the data. It was shown in [10] that, in any given trial, the precise time at which ER begins and finishes attempting to speak is not known.

4.6 Conclusions

We have obtained modest frame classification performance at levels significantly better than chance, using several firing rate estimation methods on 3 neural data sets recorded from speech motor cortex. Best performance is obtained with the histogram firing rate method, but the other firing rate methods are less susceptible to numerical instabilities. The performance of speech activity detection is better than chance, but generally poor. Our preliminary investigation shows, with modest results, that we can classify speech-related neural data according to attempted vowel production. We are motivated, with caution, to pursue our own experiments with subject ER, and to apply more powerful methods to the important next step of decoding neural data for a speech prosthesis.

CHAPTER V

DISCRETE-STATE DECODING FOR A NEURAL SPEECH PROSTHESIS

We have discussed applying Gaussian mixture classifiers to detect and classify imagined speech production activity in intracortical neural data recorded from subject ER in a previous study, the first of its kind. Decoding methods in the previous study, were based on a continuous-state approach and depend on formant frequencies of speech, suitable for vowels and some sonorant consonants. In this chapter we present a new, discrete-state approach for a neural speech prosthesis with an emphasis on articulation gestures for consonant sounds. We discuss real-time data-collection experiments we conducted with ER for our discrete-state approach designed to address issues with timing and to collect neural data of ER performing imagined bilabial, alveolar and velar articulation gestures. We use hidden Markov models on offline neural data to classify continuous segments of attempted speech recorded in these sessions.

5.1 Neural Speech Prosthesis

The ability to control speech motor function can become severely impaired or even completely disabled by traumatic brain injury and certain neurological disorders. In cases where cognitive ability remains intact, and there is evidence of remnant speech-related cortical activity, persons living with such severe speech motor disabilities may benefit from an intracortical prosthesis for speech restoration or "speech prosthesis."

As we define it, an intracortical neural speech prosthesis is composed of at least 4 major components:

1. *Implant* - An electrode surgically implanted in a region of the brain normally

related to speech production.

2. *Front-End Signal Processing* - Extracting and processing speech-related extracellular electric potentials directly from a population of cortical neurons, and quantifying neuronal firing. This includes spike detection and sorting and firing rate estimation.

3. *Decoding* - decode neuronal firing activity according to some meaningful acoustic, articulatory, phonemic or linguistic interpretation and,

4. *Speech Synthesis or Display* - produce an audio or visual output (or both) based on the decoded result.

In this chapter, we discuss our discrete-state approach to the *Decoding* stage of an intracortical neural speech prosthesis, and a series of real-time, discrete-state recording experiments we conducted with a human volunteer living with Locked-In Syndrome. The *Implant* component used in this work is the same as previously described in Section 4.1 and the *Front-End Signal Processing* component, including spike-sorting and firing rate estimation, are discussed in detail in Sections 4.2 and 4.3. Since we use a discrete-state approach, the *Speech Synthesis* component can be accomplished by displaying the decoded state sequence visually or using text-to-speech methods to produce an audio signal. We do not explicitly implement speech synthesis in this work.

5.1.1 Previous Work: Continuous-State Vowel Decoding

Our discrete-state approach to a neural speech prosthesis builds on previous work with subject ER by Brumberg, Kennedy and Guenther in decoding neural data into sustained vowel sounds [10, 28]. In these experiments, ER was presented with a joint audio-visual vowel sequence stimulus with feedback control. The audio portion of the stimulus was synthetic speech and the visual stimulus was a moving cursor on

a large screen in front of the subject. A Kalman filter was trained to decode neural firing activity in speech motor cortex into a sequence of continuous, time-varying, two-dimensional vectors $\mathbf{F} = \{\mathbf{f}_1, \ldots, \mathbf{f}_N\}$. The values of each vector $\mathbf{f}_t$ correspond to the first two resonant or formant frequencies of speech audio and are used to control both the position of the cursor on-screen as well as a formant-based speech synthesizer in real time. Each trial consisted of a Listen phase where a synthetic vowel sequence was presented to the subject and a Speak phase in which the subject was asked to repeat the same vowel sequence using the feedback-based neural interface. The study subject, ER, was able to successfully learn to control the feedback-based neural interface for production of the target vowel sequences with an average rate near 70% correct computed per trial block.

5.2 Discrete-State Decoding Framework

The decoding stage of a neural speech prosthesis has many important structural similarities to the well-known problem of automatic speech recognition (ASR). In the ASR problem, the input is a frequency-based parameterization of a speech audio signal; for a neural speech prosthesis, it is an ensemble of neural firing rate estimates. For both of these problems, the objective is to decode a multivariate, continuous-valued input signal into a time-aligned sequence of phonemic or linguistic tokens or whole words. Our decoding approach to the neural speech prosthesis problem is based on the hidden Markov model (HMM) framework, which is used extensively for acoustic modeling in ASR systems. The general hidden Markov model framework is described in detail in Section 2.2.

Given one or more channels of cortical neural activity, let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ be a sequence of instantaneous neural firing rate estimates from a population of D single- and multi-unit clusters, where each $\mathbf{x}_t$ is of length D. Let the discrete-valued sequence $\mathbf{q} = \{q_1, q_2, \ldots, q_N\}$ be a sequence of states corresponding to the intended

speech content.

Let $\mathcal{Q}$ be the complete set of possible states for the HMM. Generally, $\mathcal{Q}$ should be sufficient to represent any intended speech utterance for communication. Let $\lambda = \{\boldsymbol{\pi}, \mathbf{A}, \{\theta_j\}_{j=1}^{L}\}$ be the set of all parameters of the HMM, defining both transitions between states and the distribution of neural firing rates within each state.

The parameters in λ, including the initial state distribution $\boldsymbol{\pi}$, the state transition matrix $\mathbf{A}$, and "emission" or observation likelihood parameters $\{\theta_j\}_{j=1}^{L}$ are described in more detail in Section 2.2. As in (4) and (7), we can define the likelihood $P(\mathbf{X}; \lambda)$ of a continguous sequence of neural firing rates in $\mathbf{X}$ is expressed as follows

$$P(\mathbf{X}\ ; \lambda) = \sum_{\mathbf{q}} P(\mathbf{X}|\mathbf{q}\ ; \lambda) P(\mathbf{q}\ ; \lambda) \tag{62}$$

$$= \sum_{\mathbf{q}} \pi_{q_0} \prod_{t=1}^{N} a_{q_{t-1} q_t} p(\mathbf{x}_t\ ; \theta_{q_t}). \tag{63}$$

Based on our work in classifying short frames of neural firing rate estimates in Chapter 4, we use Gaussian mixture models for the likelihood model in each state of the HMM. As such, the observation likelihood $p(\mathbf{x}_t; \theta_i)$ for observation $\mathbf{x}_t$ in HMM state i is

$$p(\mathbf{x}\ ; \theta_i) = \sum_{k} c_k \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{64}$$

As given in (6), the likelihood of a sequence of firing rate estimates given a known or hypothesized state sequence $\mathbf{q}$ is as follows

$$P(\mathbf{X}|\mathbf{q}\ ; \lambda) = P(\mathbf{X}|\mathbf{q}\ ; \{\theta_j\}_{j=1}^{L}) = \prod_{t=1}^{N} p(\mathbf{x}_t\ ; \theta_{q_t}). \tag{65}$$

We use the Baum-Welch procedure for maximum likelihood estimation of the HMM parameters λ as described in Section 2.2.3. We decode the HMM using the Viterbi algorithm described in Section 2.2.2.

5.3 Discrete-State Recording Experiments

We have formulated a discrete-state recording paradigm explicitly designed for an HMM-based framework. The physical apparatus and hardware setup is in large part the same as described in Section 4.1. We developed a real-time, interactive software framework for conducting experiments and collecting and recording extracellular electric potentials in real time. Spike detection and sorting is done in real time according to the manual cluster cutting procedure described in Section 4.2.

In a series of data collection experiments, we presented the subject with an audio-visual stimulus consisting of a visual display on a large screen in front of him, and synchronized audio of vowel sounds, sustained consonants and silence. Figure 17 shows a screenshot of the visual portion of the stimulus. Three speech sounds auditioned for the subject correspond directly to 3 large white boxes on-screen; in this example, the speech sounds auditioned are the phonemes /ow/, /m/ and /aa/. A needle moves from left to right to precisely synchronize the task and the recorded data. In Figure 17, the position of the needle indicates that the /m/ sound is being auditioned.

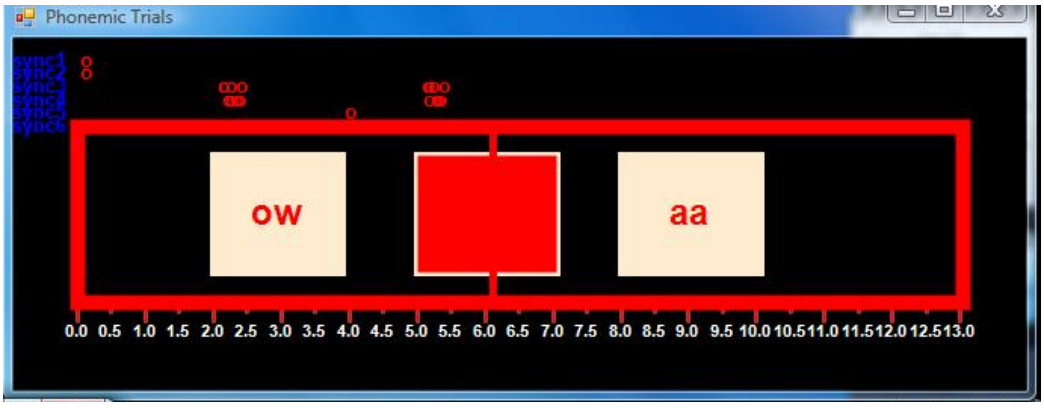


Figure 17: Screenshot of visual stimulus for discrete-state recording experiments. Elapsed time in seconds is shown toward the bottom.

5.3.1 Selection of States

Speech production is accomplished, in part, by the time-varying movement of the set of speech articulators, which includes the teeth, tongue, jaw, lips, palette and glottis (not an exhaustive list). In normal speech production, the articulators are used to modulate a air flow through the vocal tract to produce a sequence of meaningful, distinguishable sounds. Previous work outlined in Section 5.1.1 was successful in decoding speech sounds which can be synthesized based on continuous-valued resonant or formant frequencies alone. While this is sufficient for the set of vowels, as well as some sonorant consonants, a complementary approach should be applied to decode such speech sounds as obstruents, fricatives, plosives and affricates, for which no resonant frequencies are discernible.

Our HMM-based neural speech prosthesis framework is designed to decode speech-related cortical neural activity into a discrete set of states to facilitate the production of synthesized speech. Since we have chosen speech motor cortex as the implant site, the set of states should sufficiently characterize the range of speech motor function, insofar as it can be reliably discriminated from neural firing activity in the vicinity of the implant.

The set of states for our data collection experiments was carefully chosen to complement the vowel decoding work described in Section 5.1.1, and to expand the scope to include important non-vowel sounds. In Figure 18, the subset of non-vowel English phonemes of the ICSI phoneme set [46] is shown, with emphasis given to several articulatory and phonological attributes of interest.

We chose consonant phonemes for our experiments according to a set of desirable criteria given below:

1. *Audibility* - Easy for the subject to hear when auditioned in the "Listen" phase. For this reason, consonants that are voiced and naturally sustained were preferred.

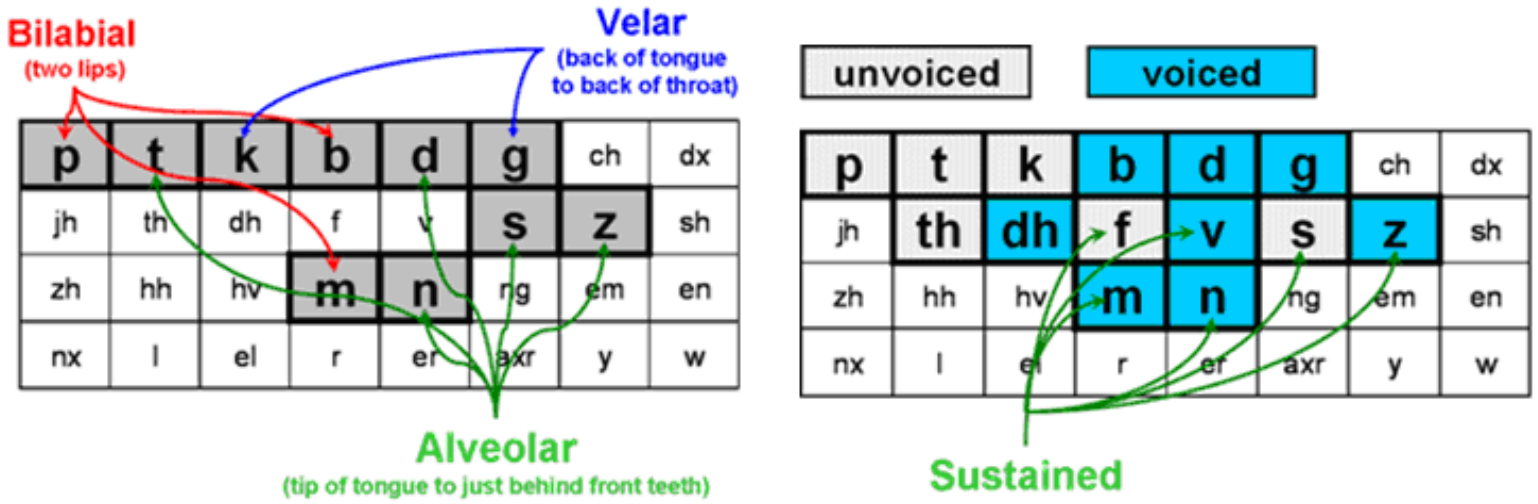


Figure 18: Articulatory and phonological attributes of selected non-vowels.

2. *Comprehensibility* - Should correspond to articulatory gestures that are easy to describe to the subject, and easy for him to imagine performing.

3. *Mutual Distinctiveness* - The set of consonants used on any given day, should correspond to a small subset of articulatory gestures which are mutually distinct from each other, in terms of the place of articulation, the articulators used or both.

5.3.2 Data Collection

We conducted 5 recording sessions for collecting data from subject ER using our real-time software framework on 12/19/2008, 1/26/2009, 2/2/2009, 2/13/2009, 2/20/2009. Each session consisted of 15 to 20, 3-phase trials. The latter 3 recording sessions were designed after the well-known Vowel-Consonant-Vowel (VCV) psychoacoustic perceptual tests used to judge the ability of listeners to recognize consonants [64]. For our purposes, we sought an experimental framework to test explicitly for discernible neural correlates of distinct articulatory gestures associated with consonant sounds.

A brief description of 18, 3-phase trials recorded on 2/20/2009 is given in Table 7. Nine "VCV" trials were conducted, each consisting of a vowel "V1," followed by a consonant "C1" and then "V1" again. These were followed by 9 similarly composed Consonant-Vowel-Consonant (CVC) trials. The consonants /m/, /n/ and /g/ are principally articulated at the lips (bilabial), the alveolar ridge (alveolar/coronal), and the velum (velar), respectively, as indicated in Table 1. The articulators used for /m/, /n/, and /g/ are the lips, the tongue tip and the tongue dorsum, also respectively. These were chosen, as described in Section 5.3.1, and illustrated in Figure 18, to cover a wide range of articulatory gestures and sounds.

Each individual trial consisted of a "Listen" phase, when the subject was instructed to listen to the audio stimulus without trying to speak, followed by a "Speak" phase where he was asked to attempt to repeat the stimulus without an audio cue.

Table 7: Description of VCV and CVC experiments.

Trial #		Vowel-Consonant-Vowel (VCV) Trials						
1	bilabial	silence	uw	silence	m	silence	uw	silence
2		silence	aa	silence	m	silence	aa	silence
3		silence	iy	silence	m	silence	iy	silence
4	alveolar	silence	uw	silence	n	silence	uw	silence
5		silence	aa	silence	n	silence	aa	silence
6		silence	iy	silence	n	silence	iy	silence
7	velar	silence	uw	silence	g	silence	uw	silence
8		silence	aa	silence	g	silence	aa	silence
9		silence	iy	silence	g	silence	iy	silence

Trial #		Consonant-Vowel-Consonant (CVC) Trials						
1	bilabial	silence	m	silence	uw	silence	m	silence
2		silence	m	silence	aa	silence	m	silence
3		silence	m	silence	iy	silence	m	silence
4	alveolar	silence	n	silence	uw	silence	n	silence
5		silence	n	silence	aa	silence	n	silence
6		silence	n	silence	iy	silence	n	silence
7	velar	silence	g	silence	uw	silence	g	silence
8		silence	g	silence	aa	silence	g	silence
9		silence	g	silence	iy	silence	g	silence

During the "Speak" phase, no sound was played for the subject. To ensure synchrony with the recording, the visual stimulus includes a vertical needle which moves from left to right, as depicted in Figure 17. In the last two sessions, we introduced an intermediate stage called "Listen+Speak," during which the subject was asked to attempt to speak synchronously with the audio stimulus. The order for the trial phases for these sessions was "Listen," "Listen+Speak" and then "Speak."

5.4 Experimental Results

We evaluate our methodology using the VCV and CVC trials recorded on 2/2/2009, 2/13/2009 and 2/20/2009, comprising 32 minutes of recorded data. These include 2-channel traces of extra-cellular electric potentials recorded at 30kHz. A new manual

cluster cutting (using the same implant) was applied by an expert for these experiments. The new cluster definitions include 40 single- and multi-units as opposed to the 56 units defined in Section 4.2 and used throughout Chapter 4.

In all of the 3-phase "VCV" and "CVC" trials conducted in the latter 3 recording sessions on 2/2/2009, 2/13/2009 and 2/20/2009, each phase (i.e., "Listen," "Listen/Speak" or "Speak") lasted approximately 13 seconds and consisted of 7 segments: 3 speech sounds and 4 short periods of silence interspersed, as shown in Figure 17 and Table 7. Overall plots of first- and second-order statistics of firing rate estimates for these 3 sessions are given in Figure 19. In all experiments in this chapter, we use only the histogram or binning firing rate estimation method with 50 ms windows spaced 10 ms apart.

We decode attempts made by our human volunteer to produce phonemes of speech by identifying a set of classes or attributes for classification or detection. The set of classes $\mathcal{Q}$ is drawn from the recording session. In a CVC trial consisting of segments $[silence, /m/, silence, /uw/, silence, /m/, silence]$, for example, the set of classes for decoding becomes $\mathcal{Q} = \{/m/, /uw/, silence\}$. Periods of silence are explicitly incorporated to model whether the subject's attempts at producing speech can be reliably discriminated from inactivity, i.e., when he is not trying.

Let $\mathbf{X}$ be the sequence of multivariate, continuous-valued firing rate estimates corresponding to a single "Listen+Speak" or "Speak" trial, lasting between 12 and 15 seconds. Let $\mathbf{X}$ be manually partitioned into M blocks of varying lengths, corresponding to segments of the recording session, such that $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_M\}$; in all experiments, $M = 7$. Although HMMs can be used to align data into segments automatically, we use them strictly as likelihood models to classify each block of data. For each block $\mathbf{X}_i$, we choose the class s_j to maximize the posterior probability $P(s_j|X_i)$. The optimal choice $\hat{s}_j$ is given by

$$\hat{s}_j = \arg\max_{s_j} \; P(\mathbf{X}_i|s_j)P(s_j). \tag{66}$$

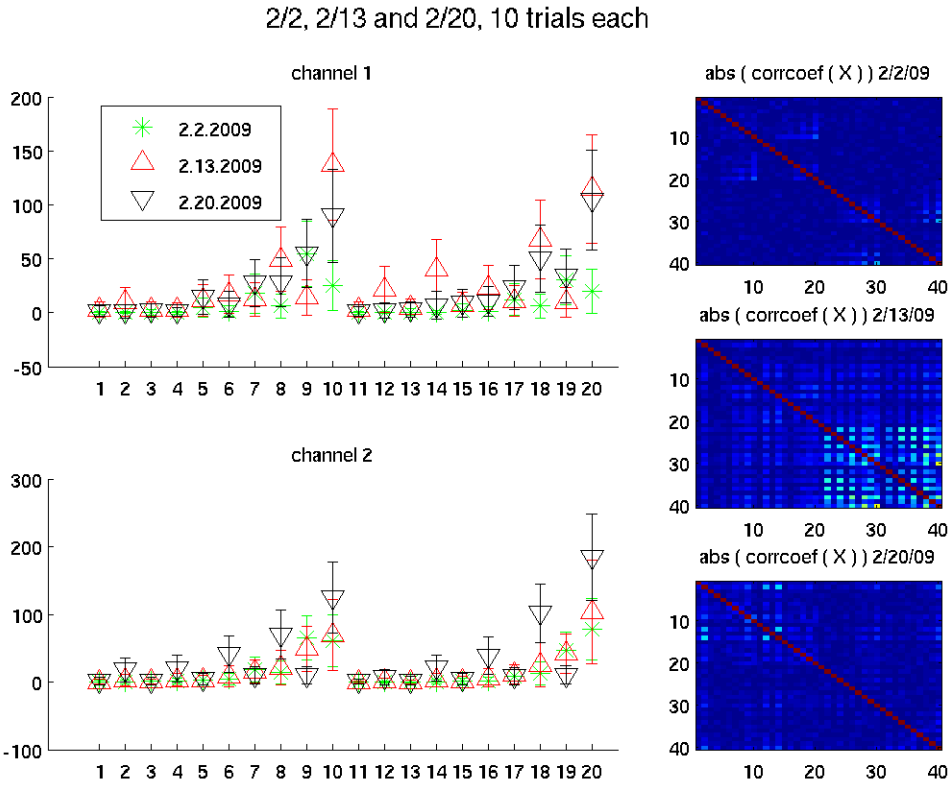


Figure 19: First- and second-order statistics of firing rate estimates. Multivariate means, standard deviations (as error bars), and magnitude correlation coefficient matrices (as images) for sessions recorded on 2/2/2009, 2/13/2009 and 2/20/2009. Statistics are collected from 10 randomly selected 12-second trials on each recording date.

For a hidden Markov model, the likelihood $P(\mathbf{X}|\mathbf{q}\ ;\lambda)$ of a hypothesized state sequence $\mathbf{q}$ is given in (65). The likelihood $P(\mathbf{X}_i|s_j)$ in (66) is for the case in which $\mathbf{q}$ is composed only of HMM states belonging to the class s_j. We will refer to a hidden Markov model with just one state as a Gaussian mixture model (GMM) when discussing classification results.

GMM and HMM classification results for 9 VCV trials recorded on 2/2/2009, consisting of a total of 63 segments, are given in Table 8. Both "Test on train" and 9-fold cross-validation results are given. A single trial was considered "perfect" if all 7 segments were classified correctly. Detailed classification results are given in

Table 8: Classification accuracy out of 63 segments for VCV trials recorded 2/2/2009.

		Train			Cross-Validation		
	Mixtures	Correct	Accuracy	Perfect	Correct	Accuracy	Perfect
GMM (1-state HMM)	1	14	0.220	0	37	0.587	1
	2	42	0.670	0	31	0.492	0
	4	49	0.780	4	32	0.508	0
	6	51	0.810	2	32	0.508	0
	8	54	0.860	3	34	0.540	0
2-state HMM	1	14	0.220	0	30	0.476	0
	2	46	0.730	0	31	0.492	0
	4	56	0.890	3	32	0.508	0
	6	57	0.900	4	29	0.460	0
	8	58	0.920	7	29	0.460	0
3-state HMM	1	14	0.220	0	30	0.476	0
	2	56	0.890	4	31	0.492	0
	4	59	0.940	6	32	0.508	0
	6	61	0.970	7	30	0.476	0
	8	61	0.970	7	33	0.524	0

Table 8 for various HMM topologies, up to 3 HMM states and 8 Gaussian mixtures per state. Performance generally increases along with increased complexity of the HMM topology, however the gap between "test on train" and generalization accuracy remains significant at all levels. Overall estimates of generalization accuracy for "Listen+Speak" and "Speak" sessions recorded on 2/2/2009, 2/13/2009 and 2/20/2009 are reported in Table 9. Classification accuracy is consistent across sessions.

Table 9: Best overall cross-validation accuracy for data recorded on 2/2/2009, 2/13/2009 and 2/20/2009.

	Model	2/2	2/13	2/20
Listen+Speak	GMM		0.571	0.571
	HMM (2-states)		0.580	0.595
	HMM (3-states)		0.571	0.571
Speak	GMM	0.540	0.563	0.595
	HMM (2-states)	0.460	0.571	0.571
	HMM (3-states)	0.524	0.571	0.571

5.5 Discussion

Our experiment protocol with subject ER involved explicitly imagining bilabial, alveolar and velar articulatory gestures. The protocol also consisted of measures to ensure timing synchrony with the data to address issues with previous experiments. Shortly after our last recording session with ER in February 2009, however, issues with severe noise in the electrode brought further recording sessions to an indefinite halt.

We evaluated our decoding methodology on 3 data sets collected in February 2009. We used our HMM framework to classify contiguous, time blocks of neural data, 1.0 to 2.0 seconds in length, into phoneme classes, rather than short 10 ms frames as in Chapter 4. We assumed knowledge of the time boundaries between blocks of data, making the task more sophisticated than simple frame classification, but less so than a full decoding task. Generally, decoding performance was found to be only slightly better than chance across data sets and various topologies of the HMM.

While our experimental protocol was designed to address timing issues in recording and to emphasize specific articulation gestures, it has at least two major limitations when compared to the continuous-state framework by Brumberg et al. [10] described in Section 5.1.1. Our protocol lacked a feedback loop. It has been shown in many neural motor control studies that subjects perform better when a feedback loop is designed into the task [86, 84]. In most of these settings however, success is determined by a continuous-valued control variable such as position or velocity. In the successful vowel-decoding experiment paradigm described in Section 5.1.1, the decoding algorithm exploits a salient, continuous-valued natural property of vowels and other sonorant speech sounds to control the output of a speech synthesizer. Using a 2-dimensional vector of speech formant frequencies as a control variable gives the subject both acoustic and visual feedback in the form of synthetic speech and the on-screen position of a cursor, respectively. For an HMM or other discrete-state decoder, audio and visual feedback information should be carefully designed into the system

to facilitate communication quickly without confusing the patient. Also, all of the sessions we conducted with subject ER (including preliminary sessions in December 2008 and January 2009) were conducted over a short 2-month period. Performing more sessions with ER over an extended would likely have led to improvements in the performance.

Further opportunities for improvement are found in the *front-end* stage of the neural prosthesis, which involves real-time signal procurement and action potential identification. In Chapter 6, we discuss some of our more recent work, where we have developed novel, rigorous methods for the task of automatic action potential identification and discrimination or "spike-sorting."

5.6 Conclusions

We conducted a series of controlled experiments with a human subject living with Locked-In Syndrome through an intracortical brain computer interface, collecting extracellular electric potentials and putative neural firing times. The experiments were specifically designed to study neural correlates of speech articulation gestures. We have also proposed a probabilistic, discrete-state decoding framework for a neural speech prosthesis based on HMMs. We have applied our HMM framework for decoding to the data collected from ER. However, due in part to a lack of feedback in our experiment protocol, we obtained results just slightly better than chance, though statistically significant. Given the known limitations of manual spike-sorting [29], we propose to improve the performance of our discrete-state framework with novel methods for spike sorting and classification.

CHAPTER VI

NOVEL METHODS FOR AUTOMATIC SPIKE-SORTING AND CLASSIFICATION

In this chapter, we present a novel, probabilistic framework for incorporating the occurrence times of neuronal action potential events or "spikes" into spike-sorting. Specifically, we jointly estimate the parameters of observed action potential waveforms and multiple hidden point processes in an iterative maximum-likelihood framework. We apply our method to two publicly available datasets of extracellular electric potentials. We then perform an empirical study of two important free parameters in our method on spike-sorting accuracy. Finally, we apply our probabilistic framework for spike-sorting to classifying short frames of vowel data for a neural speech prosthesis.

6.1 Joint Waveform and Firing Rate Spike-sorting

A number of recent studies have incorporated temporal information into spike-sorting. A complete maximum-likelihood framework based on a generalized variant of hidden Markov models (HMMs) is described in [75] to model neuronal bursting behavior. The sparse HMM framework proposed in [75] models counts of neural firing events in equally spaced time frames. In another study, extracellular traces were divided into short-duration time segments to model the non-stationarity of neuronal action potential waveform features [2]; Viterbi decoding was then used to find the optimal clustering across time segments. Several studies have explicitly incorporated models of inter-spike interval (ISI) durations into spike-sorting [70, 20]. Stationary models of spike amplitudes and ISI durations are used in [70], while HMMs are incorporated in [20] to model the time-varying firing behavior of each neuron. Finally, temporal

information was incorporated into a Bayesian network in [25] to model waveform drift and to eliminate refractory period violations. In [70, 20, 25], Markov Chain Monte Carlo distribution sampling was used for model inference.

The task of neuronal action potential identification, or "Spike-Sorting," can be seen as a latent variable problem where the set of detected firing times, and corresponding action potential waveforms are observed in an extracellular electric trace, and the identity of the underlying neurons is a hidden variable. In the remainder of this section, we describe a new approach to the Spike-Sorting problem, where we model the set of observed, threshold-crossing neuronal firing times as the aggregation of multiple hidden point processes, one for each neuron. We use an iterative procedure to estimate the maximum likelihood sequence of states based on the set of observed action potential waveforms and firing times.

6.1.1 Likelihood Model

Let the vector $\mathbf{z} = \{z_i\}_{i=1}^{N}$ be the time occurrences of N observed, threshold-crossing events corresponding to firings of a population of K cortical neuronal clusters in the vicinity of the electrode. Let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$ be the set of corresponding parameterized action potential waveforms, where each vector $\mathbf{x}_i$ has dimension D, and let $\mathbf{c}$ be an N-length, discrete-valued vector containing the set of unknown neuronal labels corresponding to each observed event.

We define the posterior probability $P(\mathbf{c}|\mathbf{X}, \mathbf{z})$ as follows:

$$P(\mathbf{c}|\mathbf{X}, \mathbf{z}) = \frac{P(\mathbf{X}, \mathbf{z}, \mathbf{c})}{P(\mathbf{X}, \mathbf{z})} \propto P(\mathbf{X}, \mathbf{z}, \mathbf{c}), \tag{67}$$

where we note that the term $P(\mathbf{X}, \mathbf{z})$ in (67) does not vary with respect to $\mathbf{c}$.

The optimal sequence $\hat{\mathbf{c}}$ thus satisfies

$$\hat{\mathbf{c}} = \arg\max_{\mathbf{c}} P(\mathbf{X}, \mathbf{z}, \mathbf{c}). \tag{68}$$

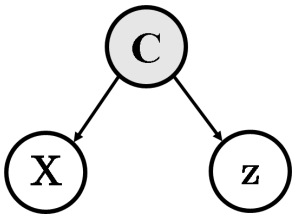


Figure 20: Statistical dependencies for parameterized waveforms $\mathbf{X}$, occurrence times $\mathbf{z}$ and labels $\mathbf{c}$.

The graphical model in Figure 20 illustrates the assumptions about the statistical dependencies between the observed variables $\mathbf{X}$ and $\mathbf{z}$, and the latent variable $\mathbf{c}$ that we will use in our modeling framework. On this basis, we express the likelihood $P(\mathbf{X}, \mathbf{z}, \mathbf{c})$ as follows:

$$\begin{aligned} P(\mathbf{X}, \mathbf{z}, \mathbf{c}) &= P(\mathbf{X}, \mathbf{z}|\mathbf{c})P(\mathbf{c}) && (69) \\ &= P(\mathbf{X}|\mathbf{c})P(\mathbf{z}|\mathbf{c})P(\mathbf{c}), && (70) \end{aligned}$$

where the terms $P(\mathbf{X}|\mathbf{c})$ and $P(\mathbf{z}|\mathbf{c})$ express the likelihood of the observed set of extracted neuronal waveforms and their corresponding occurrence times, respectively, given a sequence of neuronal labels $\mathbf{c}$, and $P(\mathbf{c})$ is the likelihood of the sequence itself.

In all experiments, we model the parameterized action potential waveform for each neuronal cluster as a single, multivariate Gaussian with parameters $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$, such that the waveform likelihood for cluster j is given by $p(\mathbf{x}; \boldsymbol{\theta}_j) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ and the likelihood for the complete set of waveforms is given by

$$p(\mathbf{X}|\mathbf{c}) = \prod_{i=1}^{N} p(\mathbf{x}_i; \boldsymbol{\theta}_{c_i}). \quad (71)$$

We characterize the temporal behavior of a population of K neuronal clusters by modeling the set of neuronal firing times $\mathbf{z}$ as the aggregation of K independent point processes $(\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_K)$, where each $\mathbf{t}_k = \{z_j\}_{j \in c_k}$ is the subset of $\mathbf{z}$ corresponding to firings of the k^{th} neuronal cluster. It is convenient to model the likelihood $P(\mathbf{t}_k)$ based on the distribution of interspike interval durations. Let $f_k(\tau; \phi_{isi})$ be a probability density function with parameter set ϕ_{isi}, characterizing the distribution of the

continuous, univariate time period $\tau = t_{k,i} - t_{k,i-1}$ between two consecutive firings of neuronal cluster k occurring at times $t_{k,i}$ and $t_{k,i-1}$. Assuming that interspike interval durations are independent and identically distributed, the likelihood $P(\mathbf{t}_k)$ can be expressed as

$$P(\mathbf{t}_k) = w_k(t_{k,1}; \phi_{init}) \prod_{i=2}^{N_k} f_k(t_{k,i} - t_{k,i-1}; \phi_{isi}) g_k(t_{k,N_k}; \psi), \tag{72}$$

where N_k is the number of neuronal firings in $\mathbf{t}_k$, $w_k(t; \phi_{init})$ is the distribution of the first firing time $t_{k,1}$, and $g_k(t_{k,N_k}) = \int_T^\infty f_k(x - t_{k,N_k}; \phi_k) dx$ is the distribution of the last firing time t_{k,N_k}, where T is the total time length of the data-set [38]. We model the likelihood $P(\mathbf{z}|\mathbf{c})$ of the complete set of firing times in terms of the joint occurrence of all class-conditional firing times, i.e., $P(\mathbf{z}|\mathbf{c}) = P(\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_K)$. Since we have assumed that these K point processes are independent, we say

$$P(\mathbf{z}|c) = P(\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_K) = \prod_{k=1}^{K} p(\mathbf{t}_k). \tag{73}$$

The last term in (70), $P(\mathbf{c})$, is the likelihood of the set of neuronal firing labels. It is important to note that since we have assumed that each neuronal cluster fires independently, all temporal modeling is expressed in terms of firing times and interspike intervals. Thus, unlike a hidden Markov model, we do not apply any explicit statistical modeling to the sequence of labels, and the likelihood $P(\mathbf{c})$ is simply given by

$$P(\mathbf{c}) = \prod_{i=1}^{N} P(c_i). \tag{74}$$

6.1.2 Clustering and Parameter Estimation

We can represent the dynamic relationship between $\mathbf{X}$, $\mathbf{c}$ and $\mathbf{z}$ with the lattice structure depicted in Figure 21. Figure 21 depicts a data set consisting of $K = 3$ neuronal clusters, with $N = 5$ observed firing times in $\mathbf{z}$ and corresponding action potential waveforms in $\mathbf{X}$. The lattice structure is similar in appearance to the commonly used

HMM trellis, but has some important differences. Particularly, since we do not explicitly model transitions between states and all temporal modeling is based on $P(\mathbf{z}|\mathbf{c})$, uneven horizontal spacing is used to illustrate observed inter-arrival durations in $\mathbf{z}$.

For the spike-sorting task, we seek to exploit both the action potential waveform shape in $\mathbf{X}$ and the temporal information in $\mathbf{z}$. To find the maximum likelihood sequence $\hat{\mathbf{c}}$ as defined in (68) we use an approximate, iterative procedure to find the best path through the state space depicted in Figure 21, similar to the well-known Viterbi procedure in HMMs. The procedure is initialized with a clustering based on the set of action potential waveforms only.

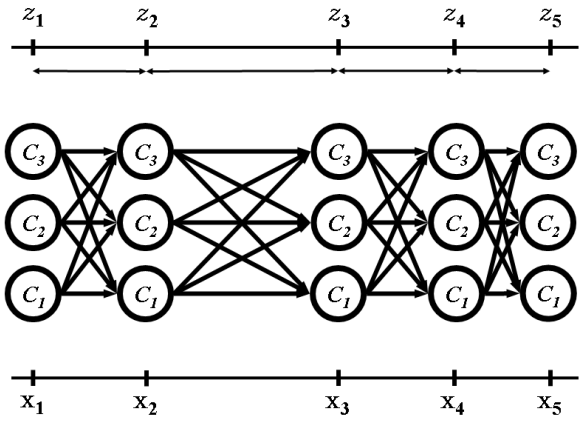


Figure 21: Lattice structure for clustering and parameter estimation.

Given a set of parameters $\lambda = \{\boldsymbol{\theta}, \boldsymbol{\phi}_{init}, \boldsymbol{\phi}_{isi}\}$, we determine the maximum likelihood state sequence $\hat{\mathbf{c}}$ by deriving a recursive expression for the joint likelihood $P(\mathbf{X}, \mathbf{z}, \mathbf{c})$. Let the notation $P(\{\mathbf{x}_i\}_{i=1}^n, \{z_i\}_{i=1}^n, \{c_i\}_{i=1}^n)$ indicate the joint likelihood of the first n data points, such that the likelihood of the full set of N data points is given by $P(\mathbf{X}, \mathbf{z}, \mathbf{c}) = P(\{\mathbf{x}_i\}_{i=1}^N, \{z_i\}_{i=1}^N, \{c_i\}_{i=1}^N)$. We decompose $P(\{\mathbf{x}_i\}_{i=1}^n, \{z_i\}_{i=1}^n, \{c_i\}_{i=1}^n)$ as follows:

$$P\left(\{\mathbf{x}_i\}_{i=1}^n, \{z_i\}_{i=1}^n, \{c_i\}_{i=1}^n\right) = P\left(\mathbf{x}_n, z_n, c_n, \{\mathbf{x}_i\}_{i=1}^{n-1}, \{z_i\}_{i=1}^{n-1}, \{c_i\}_{i=1}^{n-1}\right) \quad (75)$$

$$= P\left(\mathbf{x}_n, z_n, c_n | \{\mathbf{x}_i\}_{i=1}^{n-1}, \{z_i\}_{i=1}^{n-1}, \{c_i\}_{i=1}^{n-1}\right) \cdot P\left(\{\mathbf{x}_i\}_{i=1}^{n-1}, \{z_i\}_{i=1}^{n-1}, \{c_i\}_{i=1}^{n-1}\right) \quad (76)$$

$$= P\left(\mathbf{x}_n | c_n, \{\mathbf{x}_i\}_{i=1}^{n-1}, \{z_i\}_{i=1}^{n-1}, \{c_i\}_{i=1}^{n-1}\right) \cdot P\left(z_n | c_n, \{\mathbf{x}_i\}_{i=1}^{n-1}, \{z_i\}_{i=1}^{n-1}, \{c_i\}_{i=1}^{n-1}\right) \cdot P\left(c_n | \{\mathbf{x}_i\}_{i=1}^{n-1}, \{z_i\}_{i=1}^{n-1}, \{c_i\}_{i=1}^{n-1}\right) \cdot P\left(\{\mathbf{x}_i\}_{i=1}^{n-1}, \{z_i\}_{i=1}^{n-1}, \{c_i\}_{i=1}^{n-1}\right) \quad (77)$$

where we have used the statistical dependency assumptions illustrated in Figure 20. Assuming that $\mathbf{x}_n$ and c_n do not depend on any previous samples, we obtain

$$P\left(\{\mathbf{x}_i\}_{i=1}^n, \{z_i\}_{i=1}^n, \{c_i\}_{i=1}^{n-1}, c_n = j\right) = P\left(\mathbf{x}_n | c_n = j\right) \cdot P\left(z_n | c_n = j, \zeta_j\right) \cdot P\left(c_n = j\right) \cdot P\left(\{\mathbf{x}_i\}_{i=1}^{n-1}, \{z_i\}_{i=1}^{n-1}, \{c_i\}_{i=1}^{n-1}\right). \quad (78)$$

Note in (78) that we have expressed the likelihood of an n-length label sequence ending in state j, and that we have introduced a new variable ζ_j. Given a label sequence ending in state j, the likelihood $P(z_n|c_n = j, \zeta_j)$ depends on $\zeta_j < z_n$, which we define as the most recent, previous occurrence time of state j. To find ζ_j, some bookkeeping is necessary. Specifically, at each iteration we retain the L highest likelihood label sequences or paths through the lattice (the number of paths, L, is determined empirically). Each path contains only the most recent spikes occurring within a history time window, starting at time $z_n - \tau_{win}$ and ending at z_n. The length τ_{win} of the history window is constant, and is determined empirically. The likelihood in (78) is computed for the best L paths retained from the previous iteration. For a given path, the duration $z_n - \zeta_j$ is modeled with the inter-arrival distribution f_j. If no previous occurrences of state j are found in a given path, we say that $\zeta_j = -\infty$ and

the distribution w_j for the first firing time is used instead, with the window length τ_{win} as its argument. This is expressed in (79).

$$p(z_n|c_n = j, \zeta_j) = \begin{cases} w_j(\tau_{win}\ ; \phi_{init}) & ,\ \zeta_j = -\infty \\ f_j(z_n - \zeta_j\ ; \phi_{isi}) & ,\ \text{otherwise} \end{cases} \tag{79}$$

6.1.2.1 Iterative Procedure

Though our spike-sorting method uses both spike waveforms and firing times, we must initialize the procedure using spike waveforms only. We model the waveforms in $\mathbf{X}$ as a Gaussian mixture model (GMM), and find the maximum likelihood waveform parameters using the expectation-maximization (EM) algorithm to produce an initial clustering. Based on the initial clustering, we estimate parameters $\phi_{isi,j}$ for the inter-arrival distribution f_j, and $\phi_{init,j}$ for the first-firing distribution w_j, for each neuron j. For the first firing and inter-arrival distributions w_j and f_j, we use the exponential and lognormal probability density functions, respectively. We then assign each data point $\mathbf{x}_i$ to the maximum a posteriori GMM component to produce a clustering, and estimate parameters $\lambda = \{\theta, \phi_{isi}, \phi_{init}\}$ based on the clustering. The 3-step procedure is then:

1. Decode with parameters λ and produce a segmentation.
2. Estimate parameters λ_{new} based on the segmentation.
3. Reiterate until convergence.

6.1.3 Probability Distributions

A breakdown of the probability distributions and their parameters used in all experiments is given in Table 10. We model parameterized action potential waveforms for each cluster as single, multivariate Gaussians and model inter-arrival durations with the conditional distribution expressed in (79). For w_j, the distribution of the "first firing" after a long duration, we use a simple Poisson distribution $w_j(k;\beta t) = (\beta t)^k e^{-\beta t}/k!|_{k=1}$, with duration parameter β and event count $k = 1$. For the ISI distribution f_j, we use a log-normal density with parameters μ and σ^2. The log-normal density has been shown to have a superior empirical fit to neuronal ISI durations having a necessary minimum refractory period [70, 20].

Table 10: Breakdown of parameter set $\lambda = [\theta, \phi_{init}, \phi_{isi}]$ and probability distributions for joint waveform and firing rate spike-sorting.

Waveform	Gaussian	$\theta_j = \{\mu_j, \mathbf{\Sigma}_j\}$
First Firing $[w_j]$	Poisson	$\phi_{init,j} = \{\beta_j\}$
ISI $[f_j]$	Log-Normal	$\phi_{isi,j} = \{\mu_j, \sigma_j^2\}$

6.1.4 Parameters L and τ

In addition to the distribution parameters $\lambda = [\theta, \phi_{init}, \phi_{isi}]$, our procedure has two free parameters L and τ, which are determined empirically; these are the number of paths and the history window length, respectively. The number of paths L is typically chosen according to a trade-off of accuracy against speed and memory usage. We choose the window length τ such that the "$\zeta_j = -\infty$" condition in (79) occurs rarely. τ is chosen to be larger than an inter-arrival duration t_k for any neuron k with high probability. To estimate τ we fit a lognormal distribution to each neuronal cluster based on an intitial waveform-only clustering of the data, and choose τ_k to cover 99% of the area under the curve for neuron k. The history window τ is then simply $\tau = \max_k \tau_k$. The general expression for the log-normal density function for a variable

t with parameters μ and σ^2 is given by

$$t \sim LogNorm(t; \mu, \sigma^2) = \frac{1}{t\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\log t - \mu}{\sigma}\right)^2\right]. \tag{80}$$

Given a set of univariate, Gaussian-distributed data $x \sim \mathcal{N}(x; \mu, \sigma^2)$, if χ is the logarithm of x then, by definition, χ is log-normal distributed, such that $\chi = \log(x) \sim LogNorm(\chi; \mu, \sigma^2)$. The log-normal parameters μ and σ^2 are then the mean and variance of $\exp(\chi)$, respectively. The log-normal distribution is supported on the range $[0, \infty)$, and has been used successfully to model neuronal inter-spike interval durations [70, 20].

6.2 Experiments: Joint Framework for Spike-sorting

Given a real, continuous extracellular trace, it is typically impracticable to obtain a complete set of ground truth labels since it cannot be directly observed which neuron caused each action potential spike in the trace. This makes evaluation for spike-sorting difficult in most non-trivial cases. Synthetic extracellular traces, which are often partially composed of real data, provide fully labeled data-sets useful for development and evaluation of spike-sorting methods. When fully authentic data are desired however, it is possible to collect data using both an extracellular electrode and a carefully placed intracellular electrode in one neuronal cell to obtain a partial ground truth labeling. Spikes on an intracellular electrode identify the firing times of one neuron with near certainty. In this section, we apply our spike-sorting method to two publicly available sets of cortical extracellular traces, of which one is real and partially labeled and the other is semi-artificial and fully labeled, to demonstrate its performance.

6.2.1 WaveClus Semi-artificial Dataset

We evaluate our spike-sorting methods with labeled data collected, in part, from the publicly available WaveClus artificial data-set [72]. We use randomly selected

action potential waveforms from the "Example 1" and "Example 2" subsets, hereafter referred to as "Easy1" and "Difficult1," respectively. Each data subset consists of $K = 3$ neuronal clusters with characteristic action potential waveform shapes drawn from a library of templates. The 3 characteristic waveforms in the "Difficult1" set are similar to each other in shape and are generally more difficult to separate than in the "Easy1" set. All of the WaveClus data-sets contain realistic additive background noise at varying power levels. For our spike sorting experiments, we added additional Gaussian noise to the baseline data at various SNR levels. We use principal components analysis (PCA) for dimensionality reduction in all experiments. Scatter plots of the first 2 principal components are given in Figure 22 for all four data subsets.

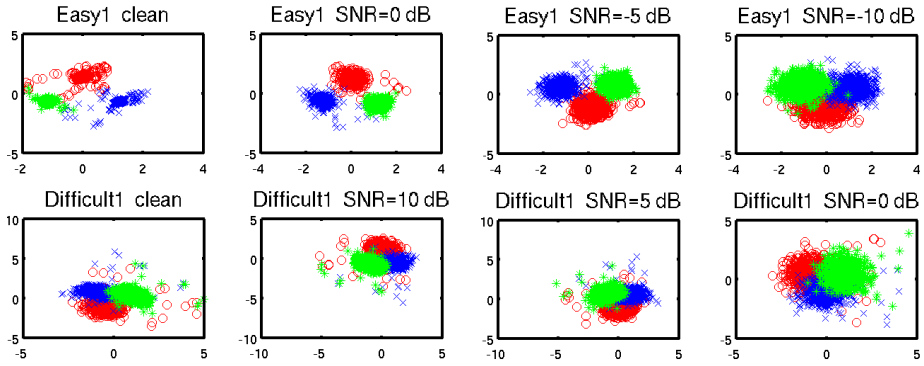


Figure 22: First 2 PCA coefficients of action potential waveforms plus noise at various SNR levels.

All subsets of the WaveClus data-set contain 3 neuronal clusters with artificial firing times having identical firing rate statistics. For our experiments, we generated firing times according to a Monte Carlo sampling of 3 independent log-normal distributions, resulting in a data-set of 2483 firing times 24 seconds in length. A minimum 3 millisecond interval duration was enforced to model the refractory period for all clusters.

Table 11 gives the simulation parameters, μ and σ^2 we used to generate inter-spike interval durations, along with the mean, in milliseconds, of the generated data. The parameters listed in Table 11 were determined by computing the sample mean

and variance of putative log inter-arrival times taken from another publicly available data-set. [1] Plots of inter-spike interval histograms are given in Figure 23.

Table 11: Simulation parameters for inter-spike interval data.

Cluster	Parameters		Mean ISI (ms)
	μ	σ^2	
1	1.5814	2.4203	24.5342
2	2.1610	1.9380	30.0564
3	1.9651	2.7068	33.9112

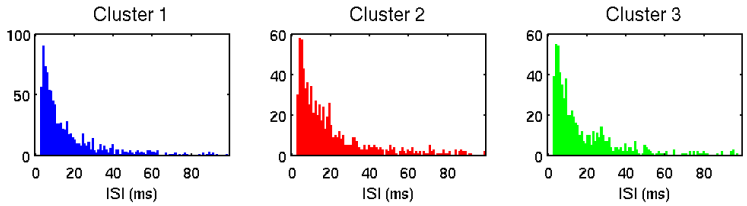


Figure 23: ISI histograms.

6.2.1.1 Results

To evaluate the accuracy of our method for spike-sorting, we compute the classification accuracy of the best match between the set of true clusters and the set of putative clusters identified by our procedure. Quantitative performance results, along with the number of iterations executed until convergence, for the WaveClus data-set are given for the baseline GMM procedure and for our joint waveform and firing rate method in Table 12. We compare our proposed approach to a GMM baseline clustering (i.e., the waveform-only initialization) and to the state-of-the-art superparamagnetic clustering or "WaveClus" method [72].

Gaussian noise was added to both the "Easy1" and "Difficult1" data-sets at various SNR levels with the original WaveClus data-set labeled "Clean" in the table. Overall,

[1] These data were collected in the Laboratory of Dario Ringach at UCLA and downloaded from the CRCNS web site.

Table 12: Classification error rates for the WaveClus semi-artificial dataset.

Dataset	SNR	GMM	WaveClus Method	Proposed
Easy1	Clean	0.93%	0.00%	0.97%
	10dB	0.72%	0.00%	0.77%
	5dB	0.89%	0.00%	0.85%
	0dB	0.81%	0.21%	0.85%
	-5dB	1.25%	0.52%	0.97%
	-10dB	8.18%	3.47%	6.20%
Difficult1	Clean	3.18%	0.45%	2.58%
	10dB	4.83%	0.94%	3.46%
	5dB	6.89%	1.36%	5.32%
	0dB	19.77%	20.0%	37.21%

we find that our method, which extends the waveform-only baseline by incorporating a hidden point process model for each neuron, reduces the error rate in the presence of noise.

The error rate for the "Easy1" data-set at high SNR levels is not significantly changed by our joint waveform and firing rate method with respect to the initial clustering, which was already quite low (less than 1.0% error). Since this data-set was particularly easy to classify, we added noise at -5dB and -10dB SNR. In the presence of high noise (-10dB SNR), we reduce the error rate from 8.18% to 6.20% by incorporating temporal information.

A similar trend is seen with the "Difficult1" data-set, but at higher SNR levels. At 10dB SNR and 5dB SNR, we reduce the error rate with respect to the baseline by 1.37% and 1.57%, respectively. However, when SNR is reduced to 0dB for the "Difficult1" data-set, we see a significant increase in the error rate. The WaveClus method however, performs significantly better on this dataset overall.

6.2.2 Continuous Extracellular Traces

To evaluate our methods on real, continuous data, we use a publicly available dataset of cortical electrical traces taken from hippocampus of anesthetized rats, hereafter

referred to as "HC1" [30]. The HC1 dataset consists of traces of extracellular (EC) electric potentials, as well as intracellular (IC) traces for 1 of K neurons in the vicinity of the EC electrodes. We use two subsets of the HC1 data-set, each 4 minutes in length, to evaluate our spike-sorting procedure. Both the EC and IC electric potential signals for Datasets 1 and 2 were recorded at a sample rate of 20 kHz. We use a highpass filter to eliminate waveform drift for the extracellular signals. A plot of a 1.79 s segment of simultaneously recorded EC and IC signals from Dataset 1 is given in Figure 24. Three peaks in the lower panel of Figure 24 indicate firing times of the "IC neuron" and correspond to 3 of the peaks in the EC signal in the upper panel.

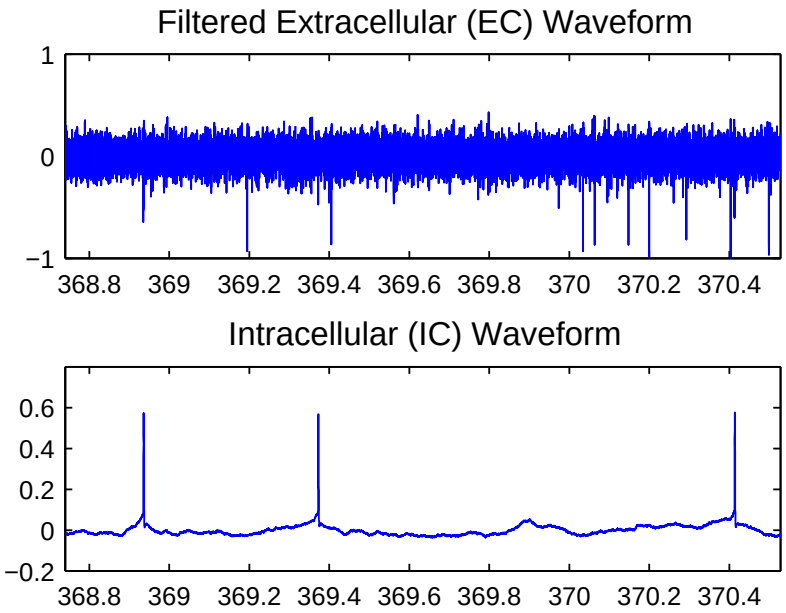


Figure 24: Extracellular (EC) and intracellular (IC) waveforms in Dataset 1 of HC1.

6.2.2.1 Methodology

We detect neuronal action potentials as "spikes" in the extracellular signal exceeding a threshold of 5σ, where σ is an estimate of the standard deviation as defined in [72]. In all spike-sorting experiments, we extract observed action potential events as 4 ms

waveforms centered at the peak point in each extracellular spike. To locate spikes on the intracellular channel in each data subset, we take the first backward difference of the IC signal and apply a peak-picking algorithm to it. EC action potentials occurring within 1 ms of an IC spike are labeled as belonging to the IC neuron. In Dataset 1, we detected 1090 total extracellular firings and 396 intracellular firings. In Dataset 2, we detect 3017 EC firings and 1100 IC firings. In each dataset, there are $K = 3$ neuronal clusters.

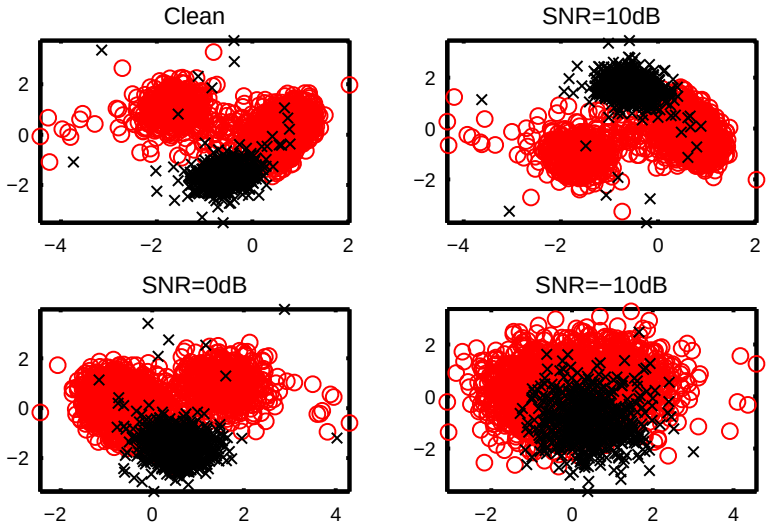


Figure 25: PCA waveform features plus noise for Dataset 2 of HC1. Features for the "IC neuron" are shown with black `'x'` markers.

For EC waveforms, we use principal components analysis (PCA) for dimensionality reduction. We keep the first 3 principal components as features for $\mathbf{X}$, the matrix of observed action potential waveforms. For our spike-sorting experiments, we add Gaussian noise to the waveform at various SNR levels before applying PCA. Scatter plots of the first two principal components at various SNR levels are given in Figure 25; extracellular waveform features corresponding to firings of the IC neuron are distinguished with black `'x'` markers.

6.2.2.2 Evaluation

Given only a partial labeling of the data, we can evaluate the performance of a spike-sorting result in terms of false positive (FP) and false negative (FN) errors for the labeled IC neuron. When a spike corresponding to the IC neuron is misclassified, a FN error is counted; inversely, when a spike is erroneously classified as belonging to the IC neuron, a FP error is counted. The error rate is defined as the sum of the FN and FP counts, divided by the number of EC firings.

6.2.2.3 Results

Quantitative performance results, in terms of the total (FP+FN) error rate are given in Table 13. For the WaveClus method, we use both a wavelet-based parameterization (the default choice for the WaveClus software package) of action potential waveforms and PCA features for a more direct comparison with the other results.

Table 13: Classification error rates (FP+FN) for the HC1 dataset.

Data-set	SNR	GMM	WaveClus Method		Proposed
			Wavelets	PCA	
Dataset 1	Clean	10.64%	32.02%	17.98%	5.60%
	15 dB	9.73%	24.5%	16.79%	4.77%
	10 dB	9.73%	31.84%	21.47%	4.95%
	5 dB	11.28%	31.93%	32.02%	6.51%
	0 dB	10.19%	32.11%	31.56%	9.27%
	-5 dB	20.09%	32.11%	32.02%	15.23%
	-10 dB	31.93%	67.98%	31.93%	32.11%
Dataset 2	Clean	2.09%	8.49%	6.56%	1.86%
	15 dB	1.96%	7.56%	6.20%	1.76%
	10 dB	1.99%	7.59%	6.27%	1.89%
	5 dB	2.29%	19.62%	11.04%	2.06%
	0 dB	3.51%	19.42%	14.29%	3.45%
	-5 dB	6.99%	19.65%	19.52%	5.87%
	-10 dB	32.55%	19.59%	19.56%	30.53%

Our proposed joint waveform and firing rate approach performs best of all but the lowest SNR level for both Datasets 1 and 2. The WaveClus method results in very

high FN error counts, but low FP counts [2]. We see a larger improvement over the GMM baseline (nearly 5% at some levels) for Dataset 1, which is ostensibly the more difficult set of the two, as evidenced by higher overall error rates for all classifiers.

6.2.3 Empirical Study of Parameters

Our clustering approach involves retaining a large number of paths, L, and a time history window of length τ of recent firings. In all of our previously reported results, we have used a fixed value of $L = 10000$ paths chosen on the basis of computational and memory constraints. The value of τ was determined to cover 99% of the area of the estimated inter-spike interval probability density curve, as described in Section 6.1.4. In this section, we perform an empirical study on the impact of our two free parameters L and τ on spike-sorting performance on the WaveClus data set. We first study the impact of increasing L on classification accuracy with τ determined as described in Section 6.1.4. Then, with a fixed value of L (we chose $L = 1000$) we study the effect of τ on the accuracy over a reasonable range. To evaluate, we simply compute the classification accuracy for the best match between the set of true clusters and the putative result.

In Figure 26, we plot the classification error rate for our spike-sorting method with the value of τ determined empirically for values of L ranging from 100 to 10000 paths on a logarithmically scaled ordinate axis. For the Easy1 data set and at higher SNR levels, the performance is largely unaffected by the number of paths L. For the Difficult1 data set, the value of L has a much more significant impact on the outcome. The impact is more pronounced for lower SNR levels, reducing the error rate for the Difficult1, 5 dB SNR case from 7.5% to 5.3% across the extremes of the range. The results in Figure 26 illustrate a trade-off of accuracy against computation and memory requirements, both of which increase with L, and suggest that except

[2] Only the total error rate is shown in Table 13

in difficult, high-noise conditions, the number of paths L can be effectively reduced with minimal impact on performance.

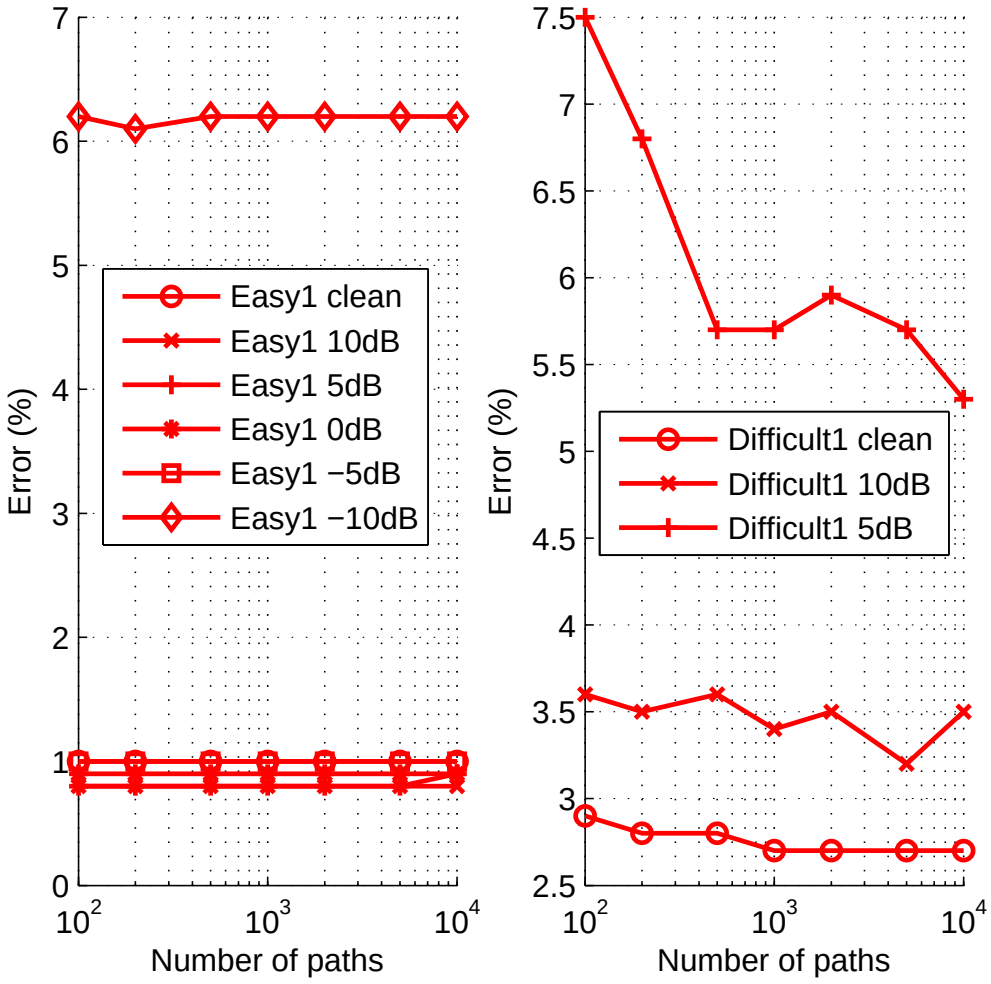


Figure 26: Error rate vs. L, the number of paths.

To evaluate the impact of the history window length τ, we apply our spike-sorting procedure to the WaveClus data over a range of values, this time holding the number of paths L fixed. Choosing a value of $L = 1000$, we implement our procedure for values of τ ranging up to 300 ms. Plots of the error rates obtained on the WaveClus data set are given in Figure 27. As with L, the accuracy is less sensitive to the value of τ in easier, low noise conditions.

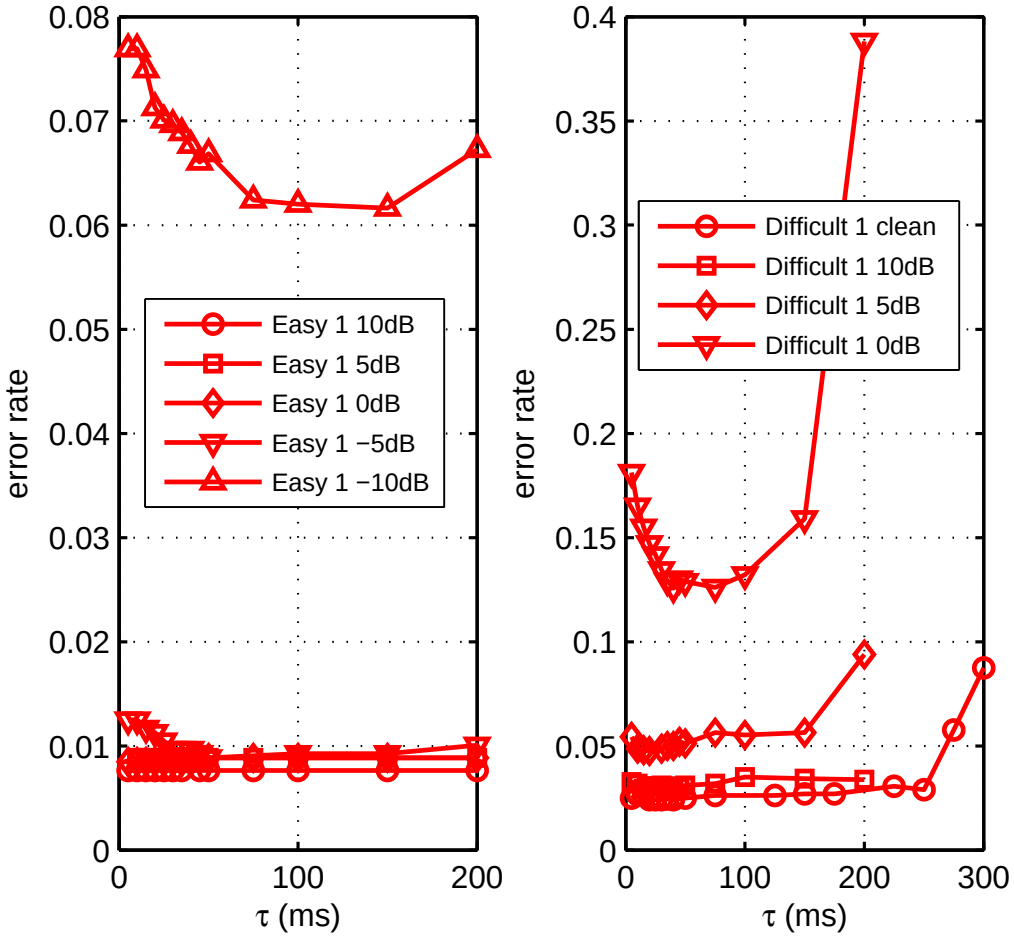


Figure 27: Error rate vs. the window length τ.

6.3 Neural Spike Classification with Discriminatively Trained Parameters

The spike-sorting task, which consists of detecting action potentials in a trace, and then assigning parameterized action potential waveforms to neuronal clusters, is generally considered an unsupervised pattern recognition problem since the clusters are determined from the observed data. Once a model of neuronal clusters has been determined, however, the task of *spike classification*, which is to classify action potentials in a brain-computer interface (BCI) based on an existing model, has its own set of important challenges. Particularly, the performance of the spike classification task is a reflection of the generalization accuracy of the model. It then follows that the spike-sorting task, in which model training or parameter estimation is usually accomplished, has a two-fold purpose: (1) identifying neuronal clusters based on action potential waveforms in set-aside data and (2) optimizing generalization performance for unseen data.

Statistical pattern recognition approaches to spike-sorting and other problems are based on the foundation of Bayes decision theory. Given an unknown observation x, a set of M classes $\{C_i\}_{i=1}^{M}$, and a statistical model of the a posteriori probability $P(C_i|x)$, for all i, Bayes decision theory prescribes that the maximum a posteriori (MAP) probability choice of C_i with respect to x minimizes the expected risk of misclassification error, assuming that all misclassifications incur an equal cost. When parametric statistical modeling approaches are used, the method of maximum likelihood (ML) is typically used for parameter estimation in $P(C_i|x)$, making the problem of designing a classifier equivalent to fitting a distribution to the data. It is important to note, however, that in order for ML methods to be optimal, it must be true that the distribution choice is adequate and that sufficient data are available for training; for automatic spike-sorting, and for other applications as well, this cannot always be

guaranteed. For this reason, alternate, *discriminative* methods of parameter estimation have been proposed for classifier design. These include the methods of maximum mutual information (MMI) and minimum classification error (MCE)[36]. Particularly, MCE is designed to minimize the misclassification risk directly.

6.3.1 Method of Minimum Classification Error

According to Bayes decision theory, the probability of error for a classification task is minimized by the maximum a posteriori (MAP) rule expressed in (81)

$$\hat{C}(\mathbf{x}_i) = \arg\max_k P(C_k|\mathbf{x}_i), \tag{81}$$

where $\{C_k\}_{k=1}^K$ is one of K classes and $\hat{C}(\mathbf{x}_i)$ is the decision made by the classifier. The method of minimum classification error (MCE) is a framework for parameter estimation that minimizes the classification error directly, rather than finding the best parametric fit to the data. The MCE method is described in the remainder of this section.

We first define the function $g_i(\mathbf{X};\theta)$, which is called the *discriminant function*. The discriminant function should be chosen such that the classifier decision is given as follows

$$C(\mathbf{X}) = \arg\max_i g_i(\mathbf{X};\theta). \tag{82}$$

It follows from (81) that $g_i(\mathbf{X};\theta)$ should be equal to the posterior probability $P(C_i|\mathbf{X})$. If the posterior probability is not available, the likelihood $P(\mathbf{X};\theta^{(i)})$ is a suitable substitution.

The decision rule in (82) is to choose class i if the value of the discriminant $g_i(\mathbf{X};\theta)$ has the greatest value among the K classes. Although this is a multinomial decision with K choices, whether or not to choose class i is a binary decision. We can formulate an approximation to this binary decision rule with a 2-step process. First we introduce a function $d_i(\mathbf{X})$ termed the *misclassification measure*. $d_i(\mathbf{X})$ should be a function of

$g_i(\mathbf{X};\theta)$ such that low values of $d_i(\mathbf{X})$ correspond to correct classifications for class i and high values of $d_i(\mathbf{X})$ correspond to misclassifications. A suitable expression for $d_i(\mathbf{X})$ is given below

$$d_i(\mathbf{X}) = -g_i(\mathbf{X};\theta) + \log\left[\frac{1}{M-1}\sum_{j,j\neq i}\exp\left[g_j(\mathbf{X};\theta)\eta\right]\right]^{1/\eta}. \tag{83}$$

Note that when η approaches ∞, the second term in (83) approaches the maximum of the value of the discriminant for all classes other than i. Next, we apply a mapping function $l(d)$ to the misclassification measure to emulate the zero-one binary decision for class i. The function $l(d)$ should be differentiable so that it is suitable for optimization methods. $l(d)$ is given below

$$l_i(\mathbf{X};\theta) = l(d_i(\mathbf{X})) \tag{84}$$

$$l(d) = \frac{1}{1+exp(-\gamma d+\rho)}, \tag{85}$$

where $l(d)$ is a sigmoid function, and $l_i(\mathbf{X};\theta)$ is called the MCE loss function. The parameters γ and ρ in (85) can be adjusted based on the data. Finally, the smoothed measure of classifier performance is

$$l(\mathbf{X};\theta) = \sum_{i=1}^{M} l_i(\mathbf{X};\theta)1(\mathbf{X}\in C_i). \tag{86}$$

With the loss function defined, it is necessary to define a procedure to find the model parameters that minimize the loss; the method of generalized probabilistic descent (GPD) is commonly used [36]. GPD is an iterative optimization technique guaranteed to converge, and to decrease the expected loss with each iteration. Let λ_j denote any model parameter for class j. The derivative $\frac{\partial l_i(\mathbf{X};\theta)}{\partial \lambda_j}$ is given in (87)

$$\frac{\partial l_i(\mathbf{X};\theta)}{\partial \lambda_j} = \frac{\partial l_i(\mathbf{X};\theta)}{\partial d_i(\mathbf{X};\theta)}\cdot\frac{\partial d_i(\mathbf{X};\theta)}{\partial g_j(\mathbf{X};\theta)}\cdot\frac{\partial g_j(\mathbf{X};\theta)}{\partial \lambda_j} \tag{87}$$

The update step for the GPD algorithm is

$$\lambda_j^{(t+1)} = \lambda_j^{(t)} - \epsilon_t \frac{\partial l_i(\mathbf{X};\theta)}{\partial \lambda_j}, \tag{88}$$

where ϵ_t, the step size, is often a constant or a simple first-order expression[36].

Expressions for the derivatives $\frac{\partial l_i(\mathbf{X};\theta)}{\partial d_i(\mathbf{X};\theta)}$ and $\frac{\partial d_i(\mathbf{X};\theta)}{\partial g_j(\mathbf{X};\theta)}$ are given in (89) and (90), respectively

$$\frac{\partial l_i(\mathbf{X};\theta)}{\partial d_i(\mathbf{X};\theta)} = \gamma \cdot l_i(\mathbf{X};\theta)[1 - l_i(\mathbf{X};\theta)] \tag{89}$$

$$\frac{\partial d_i(\mathbf{X};\theta)}{\partial g_j(\mathbf{X};\theta)} = \begin{cases} -1 & , j = i \\ \frac{\exp[g_j(\mathbf{X};\theta)]}{\sum_{s,s\neq i}\exp[g_s(\mathbf{X};\theta)]} & , j \neq i \end{cases} \tag{90}$$

6.3.2 MCE for Neural Spike Classification

Given a training set of neural action potential data, we seek to improve the classification accuracy on unseen data, assuming a parametric statistical model. In this section, we apply the MCE method to the neural classification task.

Let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ be an $N \times D$ matrix of D-length parameterized action potential waveforms for K neurons, and let $\mathbf{c}$ be a discrete-valued vector of labels. $\mathbf{c}$ can contain true labels, if they are known, or it can be the result of a spike-sorting procedure.

To apply the method of minimum classification error, we must first choose the form of the discriminant function $g_i(\mathbf{X};\theta)$. We assume a single multivariate Gaussian form for the set of parameterized action potential waveforms for each neuron and define $g_i(\mathbf{X};\theta)$ as the log likelihood $p(\mathbf{X};\theta^{(i)})$ for waveforms

$$g_i(\mathbf{X};\theta) = \log\left\{\prod_{j=1}^N \mathcal{N}(\mathbf{x}_j;\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)\right\} \tag{91}$$

$$= \log\left\{\prod_{j=1}^N \frac{1}{(2\pi)^{D/2}\left|\boldsymbol{\Sigma}_i\right|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}_j - \boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i(\mathbf{x}_j - \boldsymbol{\mu}_i)\right\}\right\}, \tag{92}$$

We define the misclassification measure $d_i(\mathbf{X})$ as expressed in (83) and the smoothed zero-one loss function as in (84) and (85). To use an optimization procedure to find the parameters θ that minimize the smoothed empirical loss function $l(\mathbf{X};\theta)$, we need to express the derivative of the smoothed loss $l_i(\mathbf{X};\theta)$ with respect to the parameters $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ for each neuronal cluster j, i.e., $\frac{\partial l_i(\mathbf{X};\theta)}{\partial \boldsymbol{\mu}_j}$ and $\frac{\partial l_i(\mathbf{X};\theta)}{\partial \boldsymbol{\Sigma}_j}$. The general chain rule expressions are given in (87), (89) and (90), with λ_j as a substitute for either $\boldsymbol{\mu}_j$ or $\boldsymbol{\Sigma}_j$. To complete the chain rule, we need to find expressions for $\frac{\partial g_j(\mathbf{X};\theta)}{\partial \boldsymbol{\mu}_j}$ and $\frac{\partial g_j(\mathbf{X};\theta)}{\partial \boldsymbol{\Sigma}_j}$. Assuming the Gaussian form in (92) with diagonal covariance matrix $\boldsymbol{\Sigma}_i$ such that $\boldsymbol{\Sigma}_i = [(\sigma_{il})^2]_{l=1}^D$, it can be shown [17] that

$$\frac{\partial g_j(\mathbf{X};\theta)}{\partial \boldsymbol{\mu}_j} = \sum_{t=1}^{N} \boldsymbol{\sigma}_j^{-1/2}(\mathbf{x}_t - \boldsymbol{\mu}_j) \tag{93}$$

and

$$\frac{\partial g_j(\mathbf{X};\theta)}{\partial \boldsymbol{\Sigma}_j} = \sum_{t=1}^{N} \left[\boldsymbol{\sigma}_j^{-1}(\mathbf{x}_t - \boldsymbol{\mu}_j)(\mathbf{x}_t - \boldsymbol{\mu}_j)^T - \mathbf{I}\right]. \tag{94}$$

Finally, the update expressions for $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ are given below

$$\boldsymbol{\mu}_j^{(t+1)} = \boldsymbol{\mu}_j^{(t)} - \epsilon_t \frac{\partial g_j(\mathbf{X};\theta)}{\partial \boldsymbol{\mu}_j} \tag{95}$$

$$\boldsymbol{\Sigma}_j^{(t+1)} = \boldsymbol{\Sigma}_j^{(t)} - \epsilon_t \frac{\partial g_j(\mathbf{X};\theta)}{\partial \boldsymbol{\Sigma}_j}. \tag{96}$$

6.4 Experiments: Discriminative Training for Spike Classification

In this section, we evaluate the method of MCE for the neural spike classification task, i.e., classifying unseen neural data given a labeled set. For simplicity, we use the semi-artificial, fully labeled WaveClus data set described previously in Section 6.2.1. The data set consists of subsets "Easy1" and "Difficult1," each containing 2483 action potential waveforms with 3 neuron classes. In all experiments, unless otherwise specified, we evaluate the generalization accuracy using a 5-fold cross-validation;

performance results are averaged across 5 randomly selected, non-overlapping cross-validation folds. We assume a Gaussian form for the neural waveforms after applying principal components analysis, as described in Section 6.3.2. In all experiments, a single Gaussian is fit to each neuron in the training data and the parameters are used as the baseline for MCE training. As a result, the form of the data and the model are reasonably well matched.

We begin by evaluating the performance of the baseline single Gaussian model and MCE training algorithm with respect to the size of the training data set. Accounting for the size of the available labeled data is important for two reasons: (1) The spike-sorting task is often done manually by an expert, thus limiting the size of the available data and (2) an advantage is expected for discriminative training methods on smaller data-sets [36].

We perform a 5-fold cross-validation limiting the training data to a randomly selected subset of sizes N=$\{$25, 75, 125, 175, 225, 275, 325, 375, 425 and 475$\}$. Results are plotted in Figure 28.

We plot classification error rate for the maximum likelihood single Gaussian model and for MCE training after 30 iterations. Results for the "Easy1" data subset are given in the upper two panels on both the training and test sets. For both training and testing, the performance of the single Gaussian and MCE models converge with more than 375 training samples. The single Gaussian model, however, performs better for smaller training set sizes. This is also observed for the "Difficult1" data set in the lower panels. The performance for the two methods converges after 125 samples in this case.

Next we evaluate the performance of MCE with respect to each iteration of the training algorithm for the full size of the crossvalidation training set of 1986 (i.e., 4/5 of 2483) data points. A plot of the total smoothed loss function $l(\mathbf{X};\theta)$ in (86), i.e., the optimization criterion for MCE, is given in the two left panels of Figure 29.

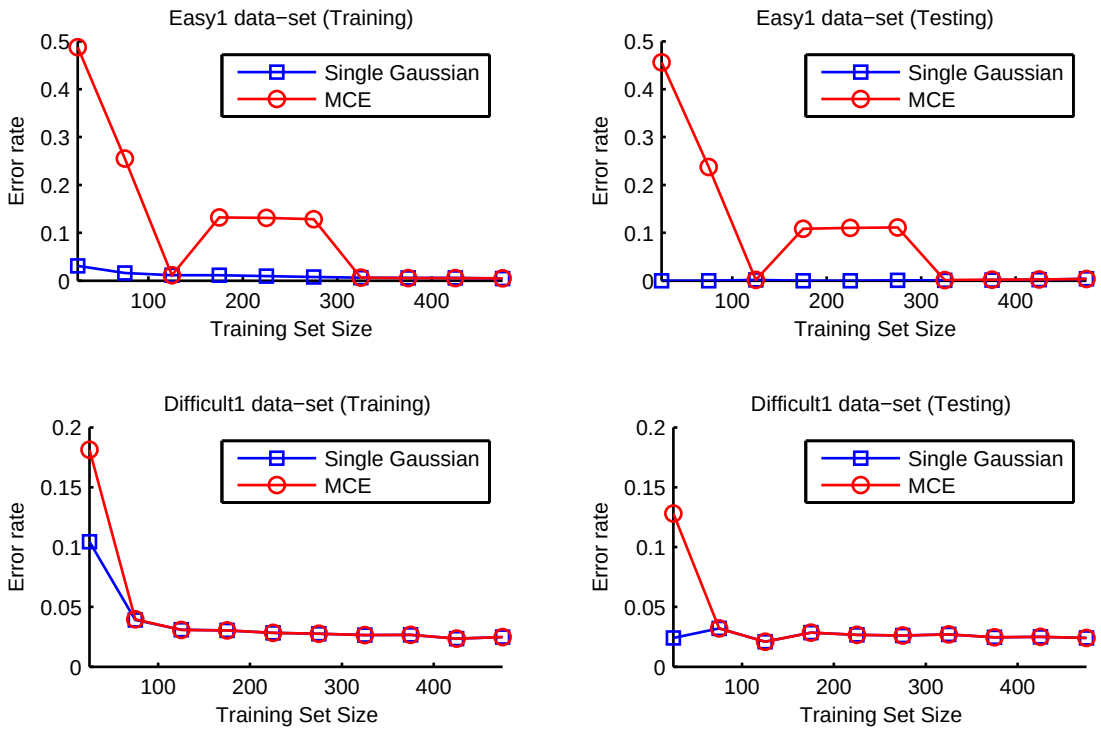


Figure 28: Average performance in terms of error rate over a 5-fold cross-validation for maximum likelihood ("Single Gaussian") and minimum classification error (MCE) methods for the Easy1 and Difficult1 subsets of the WaveClus data set.

The smoothed total loss $l(\mathbf{X};\theta)$ is plotted in the upper left and lower left panels of Figure 29 for the first crossvalidation fold (of five) of the "Easy1" and "Difficult1" data sets, respectively, for 30 training epochs. The parameters $\boldsymbol{\mu}_j$ and Σ_j are updated at each training epoch for all classes. We use the simple expression $\epsilon_t = \frac{1}{1+t}$ for the stepsize in (88), (95) and (96). For the "Easy1" dataset, the loss function decreases as designed with each iteration. For the "Difficult1" set, there is little change in the smoothed loss objective, which is near zero on the first iteration. Training and test set performance in terms of error rate are given in the two right panels in Figure 29. For both the "Easy1" and "Difficult1" cases, there is little change in both the training and test set error with each iteration.

As in Section 6.2, we study the effect of adding noise at SNR levels of 10 dB, 5 dB, 0 dB, -5 dB and -10 dB for both "Easy1" and "Difficult1." Overall training and test set performance results for the maximum likelihood single Gaussian and MCE

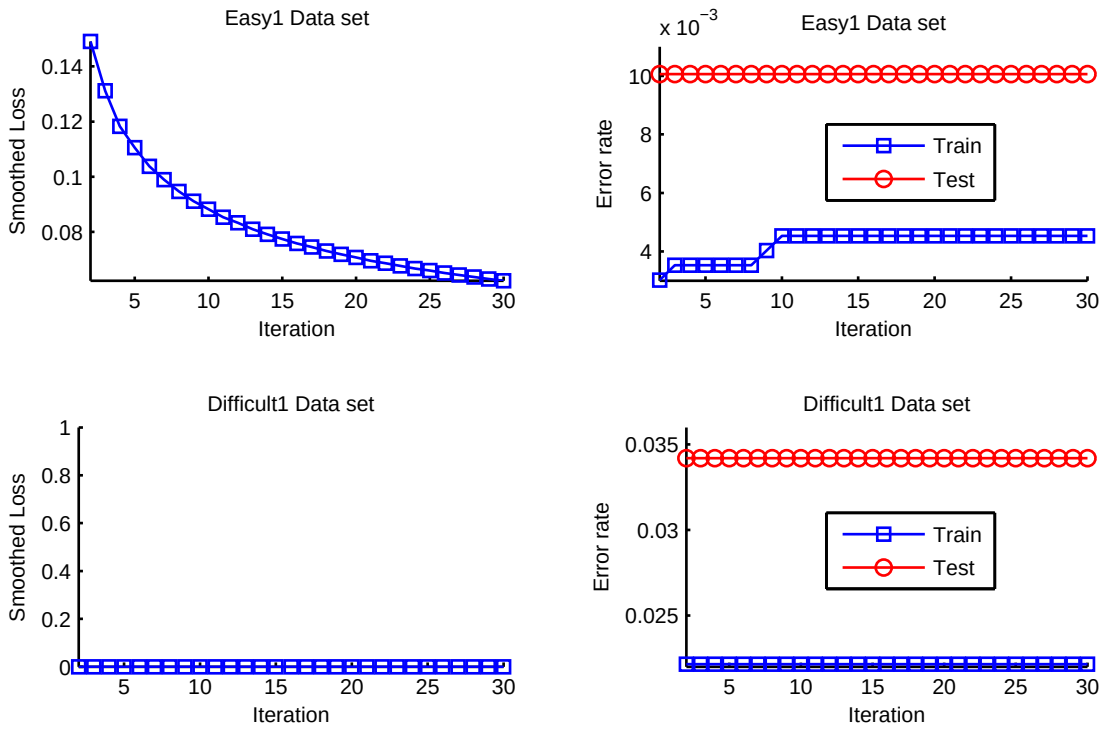


Figure 29: MCE Smoothed loss function $l(\mathbf{X};\theta)$ (left) and training and testing set error rate (right) per training iteration for the Easy1 and Difficult1 subsets of the Waveclus data set.

trained models are given in Table 14. For comparison, we also include results based on performing a spike-sorting on the training set using expectation-maximization.

For the Difficult1 data set and for the Easy1 data set at low SNR levels, we find little or no difference in performance between MCE training and the maximum likelihood single gaussian performance on the testing set. At higher SNR levels for the Easy1 data set, the error rate for ML method is lower.

Some insight into the results in Table 14 is given in Figure 30, where we plot the MCE smoothed loss function $l(\mathbf{X};\theta)$ for both data sets at all SNR levels. For the Difficult1 sets and for the Easy1 sets at lower SNR values, the smoothed loss function converges almost immediately, resulting in little or no change in classifier performances.

Table 14: Average 5-fold cross-validation performance for MCE and ML for the WaveClus data set with noise added at various SNR levels.

Data-set	SNR	Training		Testing	
		MCE	ML	MCE	ML
Easy1	clean	99.527	99.557	99.557	99.557
	10dB	99.426	99.537	99.355	99.476
	5dB	99.376	99.436	99.355	99.436
	0dB	99.235	99.245	99.234	99.275
	-5dB	99.124	99.124	99.073	99.073
	-10dB	92.549	92.549	92.428	92.428
Difficult1	clean	97.765	97.765	97.704	97.704
	10dB	97.312	97.312	97.262	97.262
	5dB	94.432	94.432	94.523	94.523
	0dB	85.753	85.753	85.986	85.986
	-5dB	69.080	69.080	69.031	69.031
	-10dB	49.335	49.335	48.934	48.934

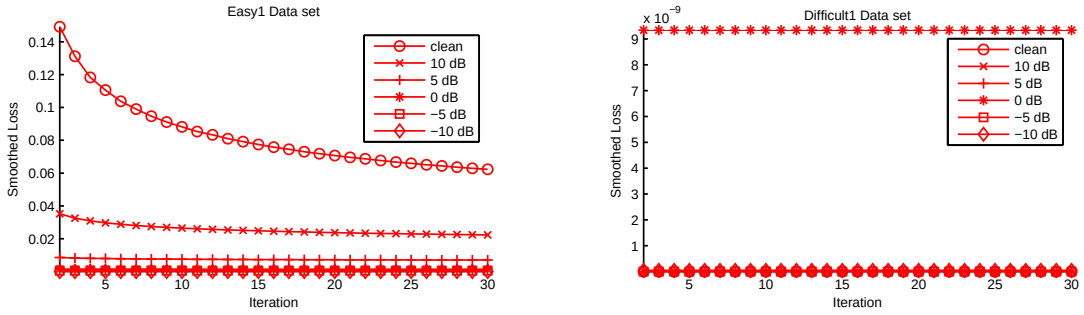


Figure 30: MCE smoothed loss function $l(\mathbf{X};\theta)$ per training iteration for the WaveClus data set various SNR levels.

6.5 Joint Waveform and Firing Rate Spike-sorting for a Neural Speech Prosthesis

In an intracortical neural prosthesis, the result of the spike-sorting operation, i.e., an ensemble of neural spike trains, is the basis of all subsequent processing. Accurate spike-sorting is important in neural prosthetic systems since errors incurred in this stage affect subsequent operations such as firing rate estimation and statistical modeling as well as the overall system performance. We have demonstrated, both with fully labeled semi-artificial data and partially labeled real extracellular traces,

that our joint waveform and firing rate framework can improve the accuracy of the spike-sorting task itself. In this section, we examine the impact of our spike sorting method on classification accuracy in the larger context of a neural speech prosthesis. We apply our joint waveform and firing rate spike clustering method to extracellular trace data collected from speech motor cortex in the context of a neural speech prosthesis as described in Chapter 4. We then use Gaussian mixture models to classify short frames of instantaneous neural firing rate estimates into intended vowels, also after the methods described in Chapter 4.

6.5.1 Selection of Neural Units

As discussed in Section 4.2, 56 single- and multi-unit neuronal clusters in extracellular data collected from subject ER were defined manually by an expert. As described in Section 6.1.2, our spike-sorting procedure is implemented by retaining a large list of hypothesized paths for a recent history of firings whose size scales with the number of neuronal clusters K. In this section, we use a statistical analysis of the manual cluster cutting result to identify a subset of these putative neural units to reduce the required computational and memory load for our spike-sorting procedure. To compare with previous frame classification results in Chapter 4, we apply Gaussian Mixture models to classify short frames of instantaneous firing rate estimates.

Based on a data set recorded on April 11, 2008, a statistical analysis [3] was performed on a all 56 putative clusters to determine (1) which neuronal clusters are significantly different from noise and (2) which clusters are likely produced by a single neuron (i.e., "single unit" clusters). First, an F-test was performed comparing the mean squared difference between the average spike waveform for each putative unit and an appropriately scaled noise signal. In this test, the mean squared difference should follow an F-distribution if the null hypothesis, i.e., that the spike waveform

[3]The statistical analysis (unpublished) was performed in 2008 by Alfonso Nieto Castañón of the Cognitive and Neural Systems Department at Boston University.

is indeed noise, is true. After adjusting for multiple testings for each neural unit, p-values from the statistical test with $p < 0.05$ indicate with confidence that the neural unit is statistically different from noise. Of the 56 putative neural units, $p < 0.05$ for 13 clusters. The cluster names, their average firing rates and p-values are listed in Table 15.

Table 15: Significance test results for 13 neural clusters with $p < 0.05$. Asterisks indicate clusters included for frame classification.

Channel	Unit	Firing Rate (Hz)	F-Test	Single Unit?
Ch1	9	32	$p = 0.000$	Y*
Ch1	17	29	$p = 0.000$	Y*
Ch1	0	30	$p = 0.000$	N
Ch1	29	21	$p = 0.000$	Y*
Ch1	8	11	$p = 0.000$	Y*
Ch1	14	10	$p = 0.001$	Y*
Ch2	5	12	$p = 0.002$	N
Ch1	21	3	$p = 0.003$	N
Ch1	5	5	$p = 0.003$	Y*
Ch1	18	3	$p = 0.005$	N
Ch1	16	11	$p = 0.011$	N
Ch2	8	3	$p = 0.030$	Y
Ch2	18	39	$p = 0.043$	N

Neuronal clusters in the data were judged to be single units or multi units, based on a simple inspection of the inter-spike interval (ISI) histogram for each cluster. It is impossible for the elapsed time between two action potential firings produced by a single neuron to be smaller than the minimum refractory period, usually on the order of milliseconds. This is reflected in the ISI histogram of single-unit clusters with the lowest valued histogram bins being empty or near-empty. In multi-unit clusters, the lowest valued histogram bins can have very high counts. Examples of ISI histograms for a single-unit cluster (Channel 1, cluster 9) and a multi-unit cluster (Channel 2, cluster 18) are given in Figure 31. In the rightmost column of Table 15, a "Y" or "N" is given to indicate which clusters were judged to be single- or multi-units.

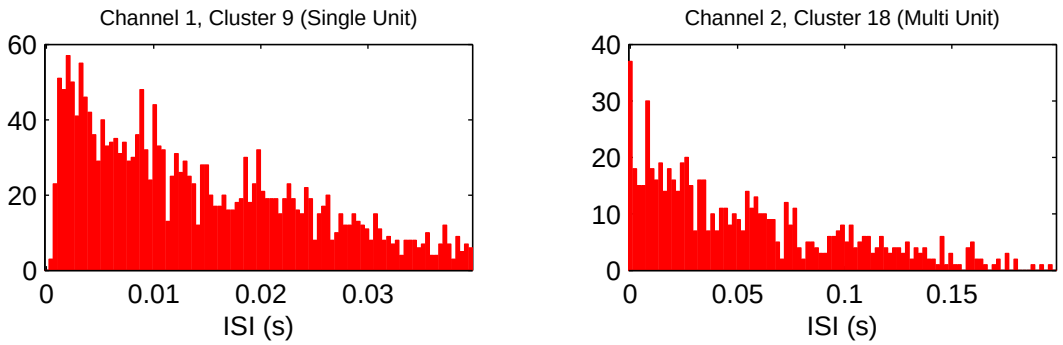


Figure 31: ISI Histograms for single- and multi-unit clusters.

We select as a subset for frame classification and spike-sorting only those clusters judged to be single units by inspection and significantly different from noise, according to the statistical test. These include Channel 1, clusters 9, 17, 29, 8, 14 and 5, and are indicated in Table 15 with asterisk (*) markers. While Channel 2, cluster 8 also meets our inclusion criteria, we exclude Channel 2 since, having just a single cluster comprises a degenerate case for the spike-sorting problem.

6.5.1.1 Frame Classification

To compare with previously obtained results in Chapter 4 we classify short duration frames of instantaneous firing rate estimates using the ensemble of spike trains corresponding to the 6 selected neuronal clusters, indicated with asterisks in Table 15. We apply the kernel smoothing method with Gaussian kernel bandwidth parameters σ=50ms (KS50), σ=100ms (KS100), σ=150ms (KS150) and σ=250ms (KS250), as well as the adaptive exponential method (AEXP) of instantaneous firing rate estimation for 10 ms frames of data. We then classify the data into a discrete set of vowel classes using Gaussian mixture model classifiers.

Vowel classification results for the Jan-11-2008, May-19-2008, and Oct-01-2008 data sets, using a 5-fold cross-validation, are given in Figure 32. Compared with previous results in Chapter 4, cross-validation error rates are generally higher using the reduced set of 6 neurons. The lowest test set error rates for the Jan-11-2008,

May-19-2008, and Oct-1-2008 data sets are 0.5976, 0.4333, and 0.4886, considerably higher than the same figures obtained with all 56 putative clusters in Chapter 4. Among all of the classification experiments, an error rate significantly better than chance is obtained only with a 16-mixture GMM classifier on the May-19-2008 data set using the the kernel smoothing firing rate method with σ=0.250 ms bandwidth.

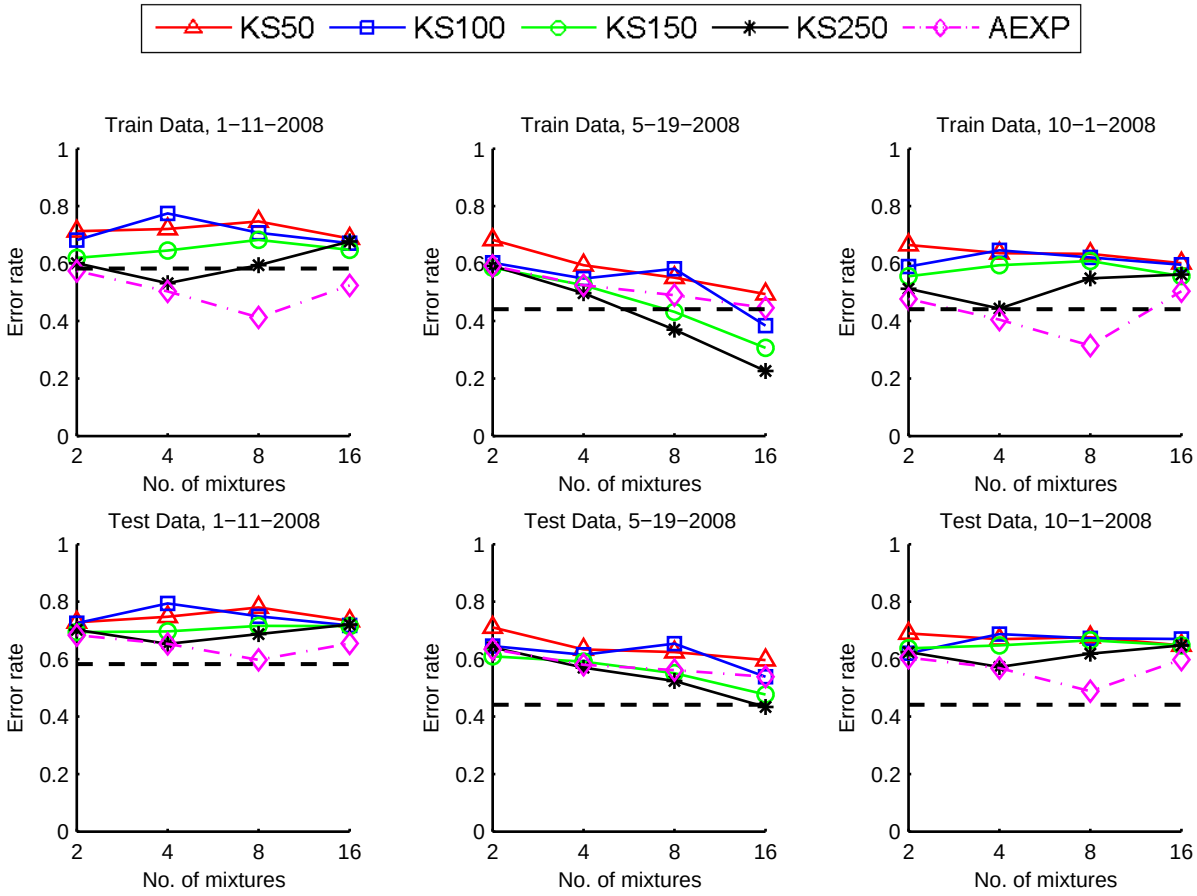


Figure 32: Frame classification error rate using 6 manually determined neuronal clusters.

6.5.2 Spike Sorting

Rather than apply our spike-sorting procedure directly to the continuous extracellular signal, we use a subset of the neural spike waveforms detected by the manual cluster-cutting procedure in Section 4.2 and then compare the results achieved by both methods. The purposes of this approach are to reduce the computational and

memory load for our procedure, to make a more direct comparison to the manual cluster cutting result, and to select, with high confidence, waveforms corresponding to neuronal firings and not noise. Let N be the total number of threshold-crossing events detected in the original manual clustering, and let K be the original number of putative clusters (56, in this case). Using the cluster cutting result, if we retain only waveforms corresponding to a subset of $k < K$ clusters, after dimensionality reduction the result $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{n}$ is a set of $n < N$ parameterized D-length waveforms, with corresponding firing times in $\mathbf{z} = \{z_i\}_{i=1}^{n}$. We then apply our spike-sorting procedure to the data in $\mathbf{X}$ and $\mathbf{z}$.

We evaluate our spike-sorting approach on the May-19-2008 data set using clusters Ch1-09, Ch1-17 and Ch1-29 only. Even with the significantly reduced set of clusters listed in Table 15, only 3 clusters was achievable with our method on the available computing resources. We use 275 seconds of data, comprising a total of 33892 detected firings. We apply principal components analysis (PCA) to the set of waveforms for dimensionality reduction and decorrelation.

In drawing comparisons between the manual cluster cutting and our proposed method for spike-sorting, at least two challenges are apparent. Perhaps the more immediate of the two is that the true labels for each neural spike are not directly observed and cannot be known with certainty. To this end, we evaluate the two clustering methods in terms of the overall system task performance for the neural prosthesis. The second major challenge is that the two clustering methods are based on very different parameterizations of the spike waveform and, as a result, can produce very different clusterings. The manual cluster cuts were determined by human inspection using the spike waveforms' peak, valley and 8^{th} sample amplitudes as features, while our proposed method uses PCA features.

A scatter plot for the manual clustering consisting of the Ch1-09, Ch1-17 and Ch1-29 clusters, using spike peak and valley amplitude, is given in the upper panel

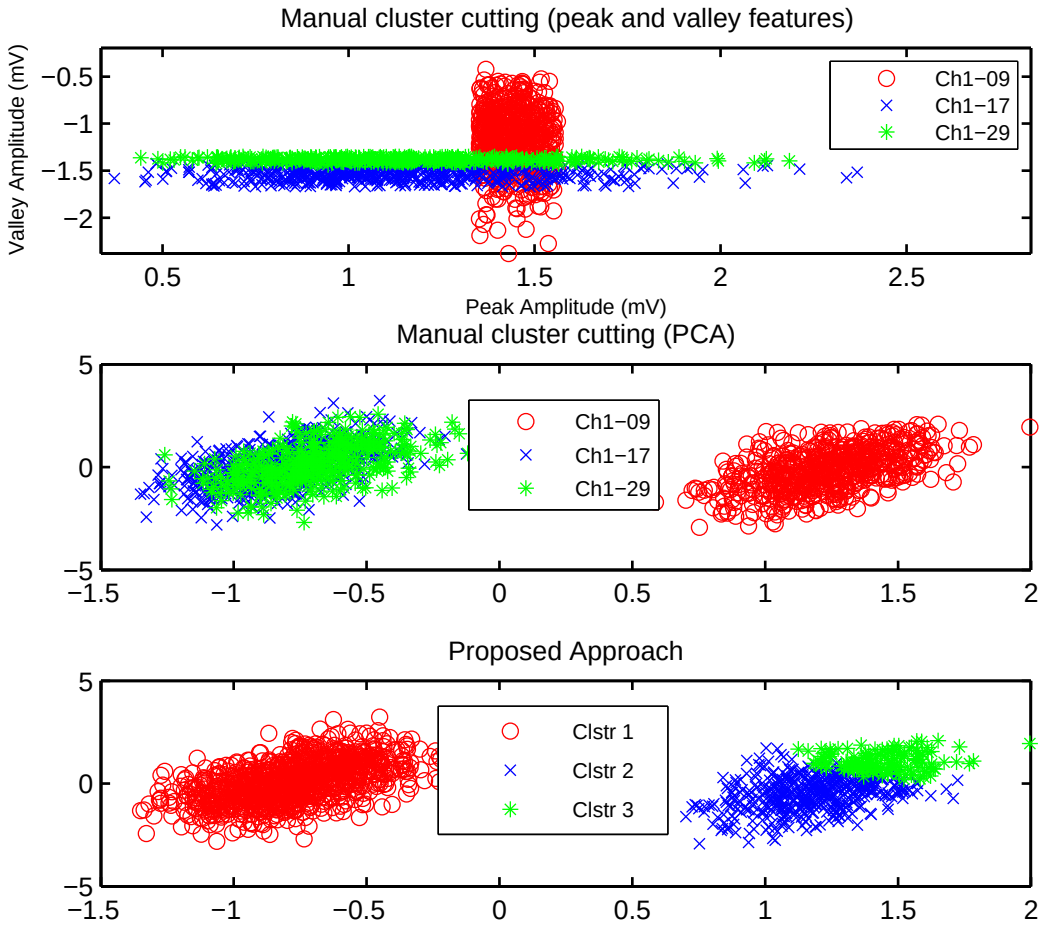


Figure 33: Waveform features for manual cluster cutting and the proposed spike-sorting approach.

of Figure 33 for 1560 neural firings taken from a single 11-second trial of the May-19-2008 data set. The same manual clustering is plotted using PCA features in the middle panel, and the spike-sorting result for our approach, using the same PCA features, is given in the lower panel. We plot both clusterings using the same features to illustrate that the two methods often produce very different results. In this case, spike data corresponding to clusters Ch1-17 and Ch1-29, were determined to belong to a single cluster by our spike-sorting method. Conversely, spikes in a single cluster (Ch1-09) correspond to two adjacent clusters as determined by our method. It should

noted that for our spike-sorting method, the number of clusters can be determined automatically or specified manually. In this case, the number of clusters was manually set to 3 to make a more direct comparison to the cluster cutting approach.

Since ground truth labels are not available for these data, we report the percent difference between the 2 clustering results by computing the error rate of the proposed method with respect to using the manual cluster cutting as a reference. For the 1560 firings depicted in Figure 33, the percent difference between clusterings is 32.5%. The average percent difference across all trials is 43.5%.

6.5.2.1 Frame Classification

Finally, using the spike-sorting result from our proposed joint waveform and firing rate method, we classify instantaneous neural firing rates into intended vowel classes as in Chapter 4 and Section 6.5.1.1. Applying the kernel smoothing and adaptive exponential methods, we compute instantaneous firing rate estimates every 10 ms. We then train GMM classifiers and evaluate classification performance using 5-fold cross-validation.

Cross-validation error rates for vowel classification on the May-19-2008 data set, based on our joint waveform and firing rate spike-sorting method, are plotted in Figure 34 for all firing rate estimation methods and GMM topologies. Frame classification performance is generally poor. Error rates are largely unaffected by the number of mixtures in the GMM classifier for all firing rate estimation methods. The lowest error rates for 2-, 4-, 8- and 16-mixture GMM classifiers are 0.6405, 0.5941, 0.5523 and 0.5649, respectively, all significantly higher than chance.

Frame classification error rates for manual cluster cutting using only the 3 clusters Ch1-09, Ch1-17 and Ch1-29 are given in Figure 35. The lowest test error rates for the 4 GMM topologies across firing rate methods are 0.6290, 0.5820, 0.5918, 0.5532. Frame classification performance, using just 3 putative neuronal clusters, is worse

than chance for both cluster cutting and for our proposed method.

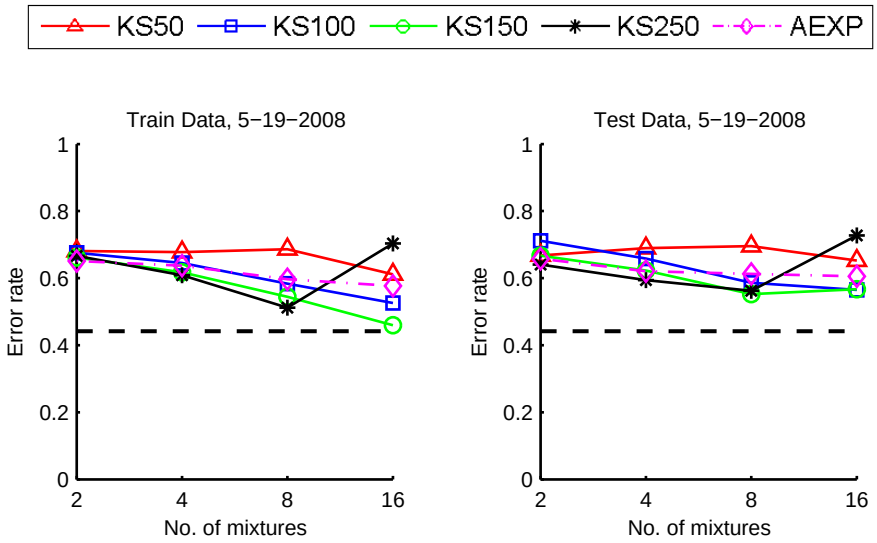


Figure 34: Frame classification error rate for the proposed spike-sorting method with 3 clusters.

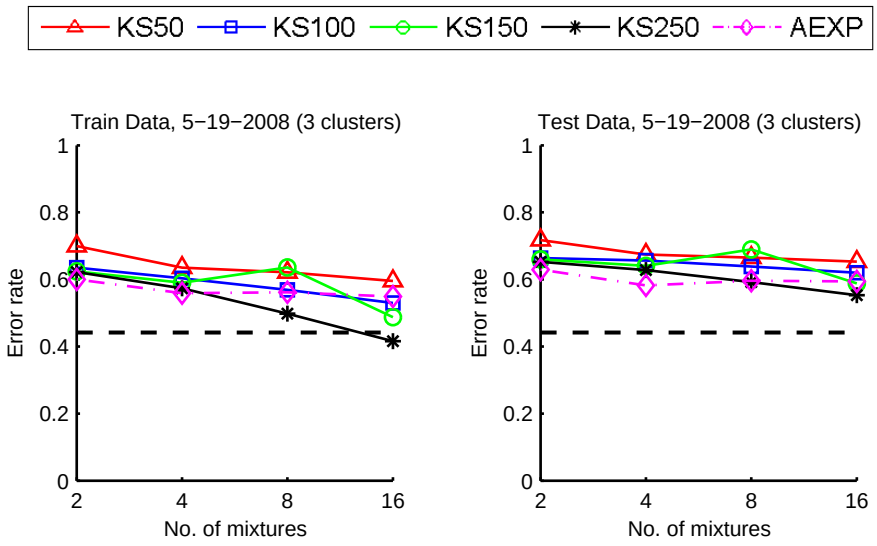


Figure 35: Frame classification error rate using 3 manually determined clusters.

6.6 Discussion

In this chapter, we proposed, implemented and tested two novel statistical approaches to the problem of spike-sorting in brain computer interfaces. The first approach, an original statistical method, is a joint probabilistic model of observed spike waveforms

and firing times. In the second approach, we apply the method of minimum classification error parameter estimation to the problem of *spike classification* on unseen data, when a set of clusters has already been defined. Finally, we apply our joint framework for spike-sorting to extracellular signals collected in the context of a neural speech prosthesis and perform frame classification on the spike-sorting result as in Chapter 4.

Our probabilistic method for spike-sorting is motivated by the idea that both the spike waveforms and their corresponding firing times constitute observed data useful for making inferences about the underlying hidden process of which neurons produced them. In doing so, we incorporate relevant data largely unused by many traditional spike-sorting approaches. We combine a single Gaussian model of spike waveforms and a first-order renewal process model of firing times for each neuron into a joint probabilistic model of several neurons in the vicinity of an electrode. The observed firing times are modeled as the aggregation of K independent point processes.

We evaluate our model on fully-labeled, semi-artificial data, as well as real, continuous, partially-labeled extracellular traces in varying noise conditions. Our method consistently performs as well as or better than a waveform-only, Gaussian mixture model clustering approach on both data sets at all noise levels. We found that our approach is especially effective in difficult, high-noise data, achieving greater improvements over the GMM baseline in these conditions. We also compare our procedure to the state-of-the-art WaveClus spike-sorting method. While the WaveClus method achieves better performance on the semi-artificial data set, our method outperforms WaveClus on a real, continuous dataset (i.e., rat hippocampus data in Section 6.2.2) by a large margin.

Our procedure for clustering and parameter estimation operates by alternately maximizing the data likelihood and estimating new parameters based on the result. While the basic idea is similar to expectation-maximization or Viterbi-based

parameter estimation in HMMs, our procedure is suboptimal with respect to the data likelihood. Since our approach depends on retaining a large number of the highest likelihood paths, performance, then, depends on the available computational resources. For this reason, we studied the impact of the number of stored paths on spike-sorting performance and found that, except in the most difficult, high-noise conditions, we could reduce the number of paths by a factor of 10 without a significant loss in accuracy.

Next we investigated whether we can improve spike classification on unseen data given an existing waveform-only, single Gaussian model using discriminative parameter estimation. The method of minimum classification error (MCE) training, was developed for pattern recognition tasks such as speech recognition where it is practically impossible to obtain a fully representative data-set for training and where the statistical model of choice (e.g., an HMM) cannot fully describe all of the variation in the data. For the spike classification task, however, the data are very well described by a simple, single Gaussian model for each neuron. It was our hypothesis that the method of MCE training could outperform maximum likelihood (ML) parameter estimation, in cases where the training set is very small. For large size training sets, we expected MCE and ML training to give comparable performance since ML estimation is optimal with sufficient training data.

We derived and implemented the method of MCE and trained and tested single Gaussian models for spike waveforms using MCE and ML estimation on the semi-artificial WaveClus dataset with varying levels of added noise and for various sizes of the training set. However, we found no advantage for MCE vs. ML. Overall classification performance for the two methods is near identical for large training sets. Where performance did differ for the two methods (i.e., for the "Easy1" data-set and for the smallest subset size of the "Difficult1" data-set) lower error rates were obtained using ML training. The results suggest that, although MCE was intended

for cases in which the training data do not adequately represent what may be seen in test data, ML training is optimal when the data and the distribution are well matched.

Finally, after showing our joint waveform and firing times spike-sorting method to be generally effective toward brain-computer interfaces with a small number of neurons, we then investigated its impact on frame classification of vowel data in a neural prosthesis for speech. Due to the computational limitations of our procedure, we use a reduced version of the data consisting of a small number of putative neurons (as determined manually by an expert) and use GMM classifiers on short frames for our spike-sorting result and the manual cluster cutting result as well. Compared to the frame classification results of Chapter 4, for which only a few configurations of the classifier and firing rate methods performed better than chance, frame classification with a small number of neurons performs significantly worse. This is true for both our proposed method, and for the manual cluster cutting result to which it was directly compared. The results are close in accuracy, but both are worse than chance. Since our spike-sorting approach is limited to BCIs with a small number of neurons, we cannot conclusively evaluate its effectiveness in a neural speech prosthesis.

6.7 Conclusions

We have demonstrated that automatic spike-sorting can benefit by using a probabilistic framework that includes action potential waveforms as well as firing times modeled as independent point processes. Our spike-sorting method outperforms a waveform-only GMM clustering approach. The improvement versus the baseline is greater in difficult and high-noise conditions. While the state-of-the-art WaveClus method achieves a lower error rate on a synthetic data-set, our method outperforms WaveClus by a large margin on real data. We determined through an empirical study

of our two free parameters, that accuracy is less sensitive to parameter tuning on easier data and in lower noise conditions. We also found that the method of minimum classification error (MCE) was not effective in reducing the error rate for the spike classification task since the data and the probability distribution used (i.e., a single Gaussian) are well matched. Finally, we were unable to show conclusively whether using our joint firing rate and waveforms spike-sorting method is more effective for improving frame classification in a neural speech prosthesis, when compared to a manual cluster cutting approach. In a comparison using a reduced set of neurons, the frame classification error rate was comparable for both methods and not better than chance for either.

We found that the method of minimum classification error (MCE) training was not effective in reducing the error rate for the spike classification task

Future work includes developing more sophisticated methods of pruning the search space for the best path.

CHAPTER VII

SUMMARY AND FUTURE WORK

7.1 Summary and Conclusions

The objective of the research we have presented was to develop probabilistic models of speech-related cortical neurological signals in the context of a brain-computer interface. The work was part of a larger study, the first of its kind, conducted with a human subject, ER, living with Locked-in Syndrome, i.e., he is both paralyzed and unable to speak. The ultimate goal of the study was to develop a neural prosthetic system to allow subject ER to control a speech synthesizer spontaneously and in real time. The study involved surgically implanting a passive microwire electrode in the subject's brain on the basis of pre-operative evidence (i.e., a functional fMRI scan) that the implant location corresponded to intended speech-related motor movement. Signals extracted from the electrode capture the extracellular electric activity of neurons at the implant location as well as other sources of noise. Laboratory equipment in the study was used to transmit and amplify signals from the electrode, detect and classify neural action potential waveforms, transmit their occurrence times to a workstation for processing and save the same data to disk.

As the prosthesis was based on speech motor function, we investigated an approach to automatic speech recognition based directly on detecting articulatory information in speech signals. Then, using neurological signals collected from a human patient, we performed speech activity detection and classification using various firing rate estimation methods. We then conducted a series of real-time data collection experiments with the patient designed explicitly to elicit speech motor activity based on bilabial,

alveolar/coronal, and velar articulatory gestures. We then used hidden Markov models to decode the data we collected offline. Next, we developed an original, probabilistic model of neuronal action potentials for spike-sorting in brain-computer interfaces. We demonstrated the method successfully using semi-artificial neural data as well as real continuous extracellular traces. We also applied this method to the neural speech prosthesis framework. Finally, we applied the method of minimum classification error to the task of neural spike classification.

We began in Section 1.1, by proposing 3 design hypothesis for the research. We repeat them here, along with the conclusions found in our investigations.

1. Evidence of articulation in speech audio signals can be used to decode intended speech content.

In this phase of the research, we aimed to detect encoded evidence of speech motor activity in speech audio signals, to study the extent to which detected speech motor gestures in a signal could be used to decode speech content. We used statistical classifiers to detect articulatory attributes of speech. Combining these and other detectors, it was shown that continuous phone recognition could be performed with only articulatory attribute scores as inputs. We have, then, demonstrated that evidence of articulation can be used to decode speech content.

2. Imagined attempts at speech activity can be decoded from neurological signals in speech motor cortex.

With neurological signals collected from speech motor cortex of a human patient during imagined speech activity, we aimed to detect, classify and decode the intended speech content from the data. The data used in Chapter 4 consisted of estimated firing times from a population of neurons in the vicinity of the electrode conducted while ER was prompted to imagine producing vowel sounds. Using a variety of firing

rate estimation methods, we obtained statistically significant results in two pattern classification tasks: frame classification and speech activity detection. Though these experiments were successful, they do not fully constitute decoding the intended speech content in the data.

We followed this with a new set of experiments in Chapter 5, which we conducted with subject ER, using sustained consonants to test explicitly for bilabial, alveolar/coronal and velar speech articulatory gestures. We then used hidden Markov models to decode the data we recorded in these experiments offline. However the results of the decoding experiments were not significantly better than chance. The experimental setup had some important limitations. Most notably, these early data collection sessions lacked a feedback loop, and no further sessions were conducted after issues with severe noise were discovered. Though we have not conclusively demonstrated a full decoding of intended speech content from cortical neural data, the speech activity detection and frame classification experiments do comprise a valuable proof-of-concept demonstration for future work toward the decoding task.

3. Temporal modeling of neural firing times can be used to improve action potential classification performance in BCIs.

Finally, we aimed to show that neural firing times could be useful toward the spike-sorting task in BCIs. To this end, we developed a new probabilistic method for automatic spike-sorting in intracortical BCIs. We modeled observed, parameterized waveforms and their observed occurrence times as part of a joint probabilistic framework, and developed an iterative method for parameter estimation. We demonstrated on two publicly available data sets, one semi-artificial and the other real, continuous and partially labeled, that our method improves spike sorting performance versus a Gaussian mixture model baseline for both data sets, and outperforms a state-of-the-art waveform-only method on the real, continuous data set. We then applied our

method to data from subject ER, but only in a limited way due to the computational limitations of the procedure. We showed that performance in stimulus decoding, in the form of frame classification, with our approach was comparable to a similarly limited version of manual spike sorting. We have, then, demonstrated that, temporal modeling of neural firing times can be used to improve spike-sorting performance in BCIs.

7.2 Summary of Thesis Contributions

In this section, we outline the contributions resulting from our research. The most significant contributions of the work were realized in the context of classifying and decoding neurological signals for a neural speech prosthesis and in developing novel methods for automatic spike sorting and classification. Our specific contributions to the state of the art are as follows:

1. *Discrete-State Neural Classification* We have conducted an evaluation of discrete-state probabilistic classification of cortical activity in speech motor cortex for a human subject during imagined speech.

2. *Experiment Protocol and Data Set* We have designed and experiment protocol for conducting precisely timed experiments in a neural speech prosthesis. We have also collected a data set of recorded extracellular traces and firing times of putative neural clusters for future research.

3. *Novel Spike-Sorting Method and Derivation* We have developed an original probabilistic method for automatic action potential classification along with mathematical derivations for the joint waveform and firing times likelihood, a recursive formulation for the likelihood in terms of previous observations and an iterative procedure for parameter estimation.

4. *Evaluation of free parameters* We have conducted an evaluation of the performance of the joint waveform and firing rate method with respect to its two free parameters, i.e., the number of retained paths and the length of the history window.

5. *Discriminative Training for Spike Waveforms* - To our knowledge, we have made the first application of minimum classification error parameter estimation for probabilistic models of neural spike waveforms.

7.3 Future Work

The larger goal of the research in Chapters 4 and 5 was to develop a real-time neural speech prosthesis for a human subject based on a discrete-state decoding approach. Though we have demonstrated parts of such a system on offline data, the completion of a real-time system remains an area for future work. Future research in this area should include the following:

- *Decoding with a Feedback Loop* - As discussed in Section 5.5, subjects in many motor control studies perform better when a feedback loop is incorporated into the system. This poses a challenge for discrete-state decoding of speech content, as control of a continuous-valued variable such as position or velocity is more easily intuited by a subject. The design of a user interface with visual or audio feedback (or both) would require intuitive displays and controls to allow a Locked-In subject to choose from a potentially large set of states (e.g., the set of English phonemes) without the use of his hands or any other extremities.

- *Speech Synthesis* - A neural prosthetic system should produce an audible speech output. With a discrete-state decoder, perhaps the simplest way is to synthesize speech from the text-based output of phonetic symbols or even whole words from the decoder. Such a system should be carefully designed to minimize potentially

large delays which could make it difficult to incorporate an audio feedback loop.

In Chapter 6, we introduced a novel method for automatic spike-sorting in intracortical brain-computer interfaces. We derived expressions for the joint likelihood and an iterative, recursive procedure for parameter estimation. Future directions for this research are briefly given below.

- *Determining K Automatically* - In all of our experiments, we initialized our procedure with a waveform-only Gaussian mixture model where the K, the number of clusters, was assumed known. Future work should include determining the number of cluster automatically. The simplest solution would be to use the Akaike information criterion (AIC) or Bayesian information criterion (BIC) for the waveform-only GMM initialization.[1] A more thorough approach might involve splitting and merging clusters on each iteration of our procedure according to AIC, BIC or some other appropriate criterion of goodness.

- *Optimal Parameter Estimation Procedure* - We use an iterative procedure to cluster the data and to determine the best parameters with respect to the likelihood. The approach is similar to Viterbi-based parameter estimation in HMMs in that it is based on finding the best path through a state-space, however our procedure is based on truncating a large list of path hypotheses and is not guaranteed to find the maximum likelihood parameters or to increase the likelihood on each iteration. Though the lattice structure in Figure 21 appears similar to the well-known HMM trellis, the first-order Markovian assumption in HMMs, i.e., that each state depends on only on the previous state, does not hold for our model, making it difficult to apply Viterbi-based parameter estimation in particular. Future work involves developing a theoretically sound, optimal procedure for clustering, guaranteed to find the best parameters in some

[1] Indeed we have done so in unpublished work and obtained similar or equivalent results.

meaningful sense.

REFERENCES

[1] Aydin, Z., Altunbasak, Y., and Borodovsky, M., "Protein secondary structure prediction for a single-sequence using hidden semi-markov models," *BMC Bioinformatics*, vol. 7, no. 1, pp. 178+, 2006.

[2] Bar-Hillel, A., Spiro, A., and Stark, E., "Spike sorting: Bayesian clustering of non-stationary data," *Journal of Neuroscience Methods*, vol. 157, pp. 303–316, October 2006.

[3] Bartels, J., Andreasen, D., Ehirim, P., Mao, H., Seibert, S., Wright, E. J., and Kennedy, P., "Neurotrophic electrode: Method of assembly and implantation into human motor speech cortex," *Journal of Neuroscience Methods*, vol. 174, pp. 168–176, September 2008.

[4] Baum, L. E., Petrie, T., Soules, G., and Weiss, N., "A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains," *Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[5] Bishop, C. M., *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[6] Brillinger, D., *Time Series: Data Analysis and Theory*. Classics in Applied Mathematics, Society for Industrial and Applied Mathematics, 2001.

[7] Bromberg, I., Fu, Q., Hou, J., Li, J., Ma, C., Matthews, B., Moreno-Daniel, A., Morris, J., Siniscalchi, S. M., Tsao, Y., and Wang, Y., "Detection-based asr in the automatic speech attribute transcription project," in *Proc. InterSpeech*, September 2007.

[8] Brown, E. N., Kass, R. E., and Mitra, P. P., "Multiple neural spike train data analysis: state-of-the-art and future challenges.," *Nature neuroscience*, vol. 7, pp. 456–461, May 2004.

[9] Brown, P. F., Della Pietra, S., Della Pietra, V. J., and Mercer, R. L., "The mathematic of statistical machine translation: Parameter estimation," *In Journal of Computational Linguistics*, vol. 19, no. 2, pp. 263–311, 1994.

[10] Brumberg, J. S., *An Electrophysiological Investigation of Human Motor Cortex and its Application to Speech Restoration.* PhD thesis, Boston University, 2009.

[11] Carlson, N. R., *Foundations of Physiological Psychology - Second Edition.* Allyn and Bacon, 2nd ed., Jan. 1992.

[12] Choi, Y.-S., Koenig, M. A., Jia, X., and Thakor, N. V., "Multiresolution entropy measure for neuronal multiunit activity," pp. 4715–4718, 2009.

[13] Chomsky, N. and Halle, M., *The Sound Pattern of English.* The MIT Press, new edition ed., Jan. 1991.

[14] Chou, W., "Discriminant-function-based minimum recognition error rate pattern-recognition approach to speech recognition," *Proceedings of the IEEE*, vol. 88, pp. 1201–1223, Aug. 2000.

[15] Clements, M. and Gavalda, M., "Voice/audio information retrieval: minimizing the need for human ears," in *Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop on*, pp. 613–623, IEEE, Dec. 2007.

[16] Cunningham, J. P., Gilja, V., Ryu, S. I., and Shenoy, K. V., "Methods for estimating neural firing rates, and their application to brainmachine interfaces," *Neural Networks*, vol. 22, pp. 1235–1246, Nov. 2009.

[17] Dan, Q., Bingxi, W., Honggang, Y., and Guannan, D., "Discriminative training of GMM based on maximum mutual information for language identification," in *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*, vol. 1, pp. 1576–1579, Oct. 2006.

[18] Davis, S. and Mermelstein, P., "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, pp. 357–366, Aug. 1980.

[19] Dayan, P. and Abbott, L., *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Computational Neuroscience, MIT Press, 2005.

[20] Delescluse, M. and Pouzat, C., "Efficient spike-sorting of multi-state neurons using inter-spike intervals information," *Journal of Neuroscience Methods*, vol. 150, pp. 16–29, January 2006.

[21] Fisher, W. M., Doddington, G. R., and Goudie-Marshall, K. M., "The DARPA speech recognition research database: Specifications and status," in *Proceedings of DARPA Workshop on Speech Recognition*, pp. 93–99, 1986.

[22] Flanagan, J., *Speech Analysis Synthesis and Perception (Kommunikation und Kybernetik in Einzeldarstellungen, 3)*. Springer-Verlag, 3rd printing ed., 1983.

[23] Foldiak, P., *The Ideal Homunculus: Statistical Inference from Neural Population Responses*, pp. 55–60. Kluwer, 1993.

[24] Ganapathiraju, A., Hamaker, J. E., and Picone, J., "Applications of support vector machines to speech recognition," *IEEE Transactions on Signal Processing*, vol. 52, pp. 2348–2355, August 2004.

[25] Gasthaus, J., Wood, F., Gorur, D., and Teh, Y. W., "Dependent dirichlet process spike sorting," in *Advances in Neural Information Processing Systems*, pp. 497–504, 2009.

[26] Gasthaus, J. A., "Spike sorting using Time-Varying dirichlet process mixture models," Master's thesis, Sept. 2008.

[27] Georgopoulos, A. P., Schwartz, A., and Kettner, R. E., "Neuronal populations coding of movement direction.," *Science*, vol. 233, pp. 1416–1419.

[28] Guenther, F. H., Brumberg, J. S., Wright, E. J., Nieto-Castanon, A., Tourville, J. A., Panko, M., Law, R., Siebert, S. A., Bartels, J. L., Andreasen, D. S., Ehirim, P., Mao, H., and Kennedy, P. R., "A wireless brain-machine interface for real-time speech synthesis," *PLoS ONE*, vol. 4, pp. e8218+, December 2009.

[29] Harris, K. D., Henze, D. A., Csicsvari, J., Hirase, H., and Buzsáki, G., "Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements," *Journal of Neurophysiology*, vol. 84, pp. 401–414, July 2000.

[30] Henze, D. A., Borhegyi, Z., Csicsvari, J., Mamiya, A., Harris, K. D., and Buzsáki, G., "Intracellular features predicted by extracellular recordings in the hippocampus in vivo," *Journal of Neurophysiology*, vol. 84, pp. 390–400, July 2000.

[31] Herbst, J. A., Gammeter, S., Ferrero, D., and Hahnloser, R. H., "Spike sorting with hidden markov models.," *Journal of neuroscience methods*, vol. 174, pp. 126–134, Sept. 2008.

[32] Hermansky, H., "Perceptual linear predictive (PLP) analysis of speech," *The Journal of the Acoustical Society of America*, vol. 87, pp. 1738–1752, Apr. 1990.

[33] Hochberg, L. R., Serruya, M. D., Friehs, G. M., Mukand, J. A., Saleh, M., Caplan, A. H., Branner, A., Chen, D., Penn, R. D., and Donoghue, J. P., "Neuronal ensemble control of prosthetic devices by a human with tetraplegia.," *Nature*, vol. 442, pp. 164–171, July 2006.

[34] Hou, J., Rabiner, L., and Dusan, S., "Automatic speech attribute transcription (ASAT) - the front end processor," in *2006 IEEE International Conference on Acoustics Speed and Signal Processing Proceedings*, pp. I–333–I–336, IEEE, 2006.

[35] Jacob, R. J. K., "The use of eye movements in human-computer interaction techniques: what you look at is what you get," *ACM Trans. Inf. Syst.*, vol. 9, pp. 152–169, April 1991.

[36] Juang, B. H., Chou, W., and Lee, C. H., "Minimum classification error rate methods for speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 5, no. 3, pp. 257–265, 1997.

[37] Juneja, A., *Speech Recognition using Acoustic Landmarks and Binary Phonetic Feature Classifiers*. PhD thesis, Department of ECE University of Maryland College Park, 2003.

[38] Kang, K. and Amari, S. I., "Discrimination with spike times and isi distributions," *Neural Comput.*, vol. 20, pp. 1411–1426, June 2008.

[39] Katagiri, S., Juang, B.-H., and Lee, C.-H., "Pattern recognition using a family of design algorithms based upon the generalized probabilistic descent method," *Proceedings of the IEEE*, vol. 86, pp. 2345–2373, Aug. 2002.

[40] Kemere, C., Santhanam, G., Yu, B. M., Ryu, S., Meng, T., and Shenoy, K. V., "Model-based decoding of reaching movements for prosthetic systems,"

in *Proc. Intl. Conf. of the IEEE Engineering in Medicine and Biology Society*, pp. 4524–4528, Sept. 2004.

[41] Kemere, C., Santhanam, G., Yu, B. M., Afshar, A., Ryu, S. I., Meng, T. H., and Shenoy, K. V., "Detecting neural-state transitions using hidden markov models for motor cortical prostheses," *J Neurophysiol*, vol. 100, pp. 2441–2452, October 2008.

[42] Kennedy, P. R., "The cone electrode: a long-term electrode that records from neurites grown onto its recording surface.," *Journal of neuroscience methods*, vol. 29, pp. 181–193, September 1989.

[43] Kennedy, P. R., Bakay, R. A. E., Moore, M. M., Adams, K., and Goldwaithe, J., "Direct control of a computer from the human central nervous system," *Rehabilitation Engineering, IEEE Transactions on*, vol. 8, pp. 198–202, August 2002.

[44] Kennedy, P. R., Kirby, M. T., Moore, M. M., King, B., and Mallory, A., "Computer control using human intracortical local field potentials," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on [see also IEEE Trans. on Rehabilitation Engineering]*, vol. 12, no. 3, pp. 339–344, 2004.

[45] Kim, K. H. and Kim, S. J., "Neural spike sorting under nearly 0-dB signal-to-noise ratio using nonlinear energy operator and artificial neural-network classifier," *Biomedical Engineering, IEEE Transactions on*, vol. 47, pp. 1406–1411, Oct. 2000.

[46] Kirchhoff, K., *Robust Speech Recognition Using Articulatory Information*. PhD thesis, University of Bielefeld, 1999.

[47] Krusienski, D. J., Sellers, E. W., Cabestaing, F., Bayoudh, S., McFarland, D. J., Vaughan, T. M., and Wolpaw, J. R., "A comparison of

classification techniques for the p300 speller," *Journal of Neural Engineering*, vol. 3, pp. 299–305, Dec. 2006.

[48] Kwon, K. Y., Eldawlatly, S., and Oweiss, K. G., "NeuroQuest: A comprehensive tool for large scale neural data processing and analysis," pp. 622–625, Apr. 2009.

[49] Le Roux, J. and McDermott, E., "Optimization methods for discriminative training," in *International Conference on Spoken Language Processing (ICSLP)*, pp. 3341–3344, Sept. 2005.

[50] Levinson, S. E., "Continuously variable duration hidden markov models for automatic speech recognition," *Comput. Speech Lang.*, vol. 1, pp. 29–45, March 1986.

[51] Lewicki, M. S., "A review of methods for spike sorting: the detection and classification of neural action potentials.," *Network (Bristol, England)*, vol. 9, November 1998.

[52] Li, J., Tsao, Y., and Lee, C. H., "A study on knowledge source integration for candidate rescoring in automatic speech recognition," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 837–840, 2005.

[53] Linderman, M. D., Santhanam, G., Kemere, C. T., Gilja, V., O'Driscoll, S., Yu, B. M., Afshar, A., Ryu, S. I., Shenoy, K. V., and Meng, T. H., "Signal processing challenges for neural prostheses," *Signal Processing Magazine, IEEE*, vol. 25, pp. 18–28, December 2007.

[54] Mansjur, D. S. and Juang, B. H., "Incremental learning of mixture models for simultaneous estimation of class distribution and inter-class decision boundaries," in *2008 19th International Conference on Pattern Recognition*, pp. 1–4, IEEE, Dec. 2008.

[55] Matthews, B. and Clements, M., "Joint modeling of observed inter-arrival times and waveform data with multiple hidden states for neural spike-sorting," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 637–640, IEEE, May 2011.

[56] Matthews, B. A., Brumberg, J., Kim, J., Clements, M. A., Kennedy, P. R., Geunther, F., Wright, E. J., Siebert, S., Bartels, J., Andreasen, D., and Nieto-Castanan, A., "Automatic detection of speech activity from neural signals in motor speech area," in *Society for Neuroscience Meeting 2008*, November 2008.

[57] Matthews, B., Chaudhari, U., and Ramabhadran, B., "Fast audio search using vector space modelling," in *2007 IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, pp. 641–646, IEEE, December 2007.

[58] Matthews, B. and Clements, M., "Joint waveform and firing rate Spike-Sorting for continuous extracellular traces," in *Forty-Fifth Asilomar Conference on Signals, Systems and Computers*, Nov. 2011.

[59] Matthews, B., Kim, J., Brumberg, J., and Clements, M., "A probabilistic decoding approach to a neural prosthesis for speech," in *International Conference on Bioinformatics and Bioengineering*, 2010.

[60] Miller, E. K. and Rainer, G., "Neural ensemble states in prefrontal cortex identified using a hidden markov model with a modified em algorithm," *Neural-computing*, vol. 32-33, pp. 961–966, 2000.

[61] Morris, J. and Fosler-Lussier, E., "Combining phonetic attributes using conditional random ?elds," in *Proc. InterSpeech-06*, 2006.

[62] Nilsson, D. and Goldberger, J., "Sequentially finding the n-best list in hidden markov models," in *Seventeenth International Joint Conference on Artificial Intelligence*, 2001.

[63] Obeid, I. and Wolf, P. D., "Evaluation of spike-detection algorithms fora brain-machine interface application," *Biomedical Engineering, IEEE Transactions on*, vol. 51, pp. 905–911, June 2004.

[64] Ohman, S. E., "Coarticulation in vcv utterances: spectrographic measurements.," *The Journal of the Acoustical Society of America*, vol. 39, pp. 151–168, January 1966.

[65] Ostendorf, M., Kannan, A., Austin, S., Kimball, O., Schwartz, R., and Rohlicek, J. R., "Integration of diverse recognition methodologies through reevaluation of n-best sentence hypotheses," in *HLT '91: Proceedings of the workshop on Speech and Natural Language*, (Morristown, NJ, USA), pp. 83–87, Association for Computational Linguistics, 1991.

[66] Oweiss, K. G., Shetliffe, M. M., and Eldawlatly, S., "Revamping signal processing for adaptive, real time, bi-directional brain machine interface systems," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pp. 5197–5200, 2008.

[67] Penfield, W. and Roberts, L., *Speech and Brain-Mechanisms*. Princeton, NJ: Princeton University Press, 1 ed., 1959.

[68] Platt, J. C., *Advances in Large Margin Classifiers*, ch. Probabilities for SV Machines. MIT Press, 1999.

[69] Plum, F. and Posner, J., *The Diagnosis of Stupor and Coma*. Contemporary neurology series, F. A. Davis, 1978.

[70] Pouzat, C., Delescluse, M., Viot, P., and Diebolt, J., "Improved Spike-Sorting By Modeling Firing Statistics and Burst-Dependent Spike Amplitude Attenuation: A Markov Chain Monte Carlo Approach," *J Neurophysiol*, vol. 91, pp. 2910–2928, June 2004.

[71] Povey, D., Kingsbury, B., Mangu, L., Saon, G., Soltau, H., and Zweig, G., "fMPE: Discriminatively trained features for speech recognition," in *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, vol. 1, pp. 961–964, 2005.

[72] Quiroga, R. Q., Nadasdy, Z., and Shaul, Y. B., "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *Neural Comput.*, vol. 16, pp. 1661–1687, August 2004.

[73] Rabiner, L. R., "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257–286, Aug. 1989.

[74] Radons, G., Becker, J., Dülfer, B., and Krüger, J., "Analysis, classification, and coding of multielectrode spike trains with hidden markov models," *Biological Cybernetics*, vol. 71, pp. 359–373, August 1994.

[75] Sahani, M., *Latent Variable Models for Neural Data Analysis*. PhD thesis, California Institute of Technology, May 1999.

[76] Santhanam, G., Sahani, M., Ryu, S. I., and Shenoy, K. V., "An extensible infrastructure for fully automated spike sorting during online experiments," in *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 4380–4384, IEEE, 2004.

[77] Schmidt, E. M., "Computer separation of multi-unit neuroelectric data: a review.," *Journal of neuroscience methods*, vol. 12, pp. 95–111, Dec. 1984.

[78] Serruya, M. D., Hatsopoulos, N. G., Paninski, L., Fellows, M. R., and Donoghue, J. P., "Instant neural control of a movement signal.," *Nature*, vol. 416, pp. 141–142, March 2002.

[79] Shimodaira, H., Okui, J., and Nakai, M., "Improving the generalization performance of the MCE/GPD LEARNING," in *Proceedings of ICSLP-98*, 1998.

[80] Shoham, S., "Robust, automatic spike sorting using mixtures of multivariate t-distributions," *Journal of Neuroscience Methods*, vol. 127, pp. 111–122, Aug. 2003.

[81] Siniscalchi, S. M., Li, J., and Lee, C.-H., "A study on lattice rescoring with knowledge scores for automatic speech recognition," in *Proc. InterSpeech-06*, 2006.

[82] Suhail, Y. and Oweiss, K. G., "Multiresolution bayesian detection of multiunit extracellular spike waveforms in multichannel neuronal recordings," in *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pp. 141–144, IEEE, 2005.

[83] Takahashi, S., Anzai, Y., and Sakurai, Y., "Automatic sorting for multineuronal activity recorded with tetrodes in the presence of overlapping spikes.," *Journal of neurophysiology*, vol. 89, pp. 2245–2258, Apr. 2003.

[84] Taylor, D. M., Tillery, S. I., and Schwartz, A. B., "Direct cortical control of 3D neuroprosthetic devices," *Science*, vol. 296, no. 5574, 2002.

[85] Thorbergsson, P. T., Jorntell, H., Bengtsson, F., Garwicz, M., Schouenborg, J., and Johansson, A. J., "Spike library based simulator for

extracellular single unit neuronal signals," in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pp. 6998–7001, 2009.

[86] Velliste, M., Perel, S., Spalding, M. C., Whitford, A. S., and Schwartz, A. B., "Cortical control of a prosthetic arm for self-feeding," *Nature*, vol. 453, pp. 1098–1101, May 2008.

[87] Ventura, V. A., "Traditional waveform based spike sorting yields biased rate code estimates.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 106, pp. 6921–6926, Apr. 2009.

[88] Victor, J. D., "Binless strategies for estimation of information from neural data," *Physical Review E*, vol. 66, pp. 051903+, Nov. 2002.

[89] Wang, Y. and Fosler-Lussier, E., "Integrating phonetic boundary discrimination explicitly into hmm systems," in *Proc. InterSpeech-06*, 2006.

[90] Wilpon, J. G., Rabiner, L. R., Lee, C. H., and Goldman, E. R., "Automatic recognition of keywords in unconstrained speech using hidden markov models," *Acoustics, Speech, and Signal Processing, IEEE Transactions on*, vol. 38, no. 11, pp. 1870–1878, 1990.

[91] Wood, F. and Black, M., "A nonparametric bayesian alternative to spike sorting," *Journal of Neuroscience Methods*, vol. 173, pp. 1–12, Aug. 2008.

[92] Wu, W., Black, M. J., Gao, Y., Bienenstock, E., Serruya, M., and Donoghue, J. P., "Inferring hand motion from multi-cell recordings in motor cortex using a kalman filter," in *SAB'02-Workshop on Motor Control in Humans and Robots: On the Interplay of Real Brains and Artificial Devices*, pp. 66–73, 2002.

[93] Wu, W., Black, M. J., Gao, Y., Bienenstock, E., Serruya, M., Shaikhouni, A., and Donoghue, J. P., "Neural decoding of cursor motion using a kalman filter," in *Advances in Neural Information Processing Systems 15* (Becker, S., Thrun, S., and Obermayer, K., eds.), pp. 133–140, MIT Press, 2003.

[94] Wu, W., Gao, Y., Bienenstock, E., Donoghue, J. P., and Black, M. J., "Bayesian population coding of motor cortical activity using a kalman filter.," *Neural Computation*, vol. 18, pp. 80–118, 2005.

[95] Yamada, T. and Meng, E., *Practical Guide for Clinical Neurophysiologic Testing: EEG*. Lippincott Williams & Wilkins, pap/psc ed., Nov. 2009.

[96] Yu, S., "Hidden semi-markov models," *Artificial Intelligence*, vol. 174, pp. 215–243, February 2010.

[97] Zen, H., Tokuda, K., Masuko, T., Kobayashi, T., and Kitamura, T., "Hidden semi-markov model based speech synthesis," in *Proc. ICSLP*, pp. 1397–1400, 2004.

VITA

Brett Alexander Matthews was born in the borough of Brooklyn in the City of New York on August 15, 1978. Mr. Matthews attended the Brooklyn Technical High School in the Fort Greene section of Brooklyn, graduating in 1996. He went on to attend Rensselaer Polytechnic Institute in Troy, NY, obtaining a Bachelor of Science degree in Computer & Systems Engineering in 2001, with honors.

Mr. Matthews then attended Georgia Institute of Technology in Atlanta, GA for graduate study, obtaining the Master of Science degree in 2003, and the Doctor of Philosophy degree in 2012, both in Electrical and Computer Engineering (ECE). While at Georgia Tech, he was awarded numerous fellowships and scholarships including the GEM Fellowship, the Student Teacher Enhancement Partnership (STEP) fellowship, the IBM FOCUS Fellowship award, the Georgia Tech and Emory University Technological Innovation: Generating Economic Results (TI:GER) Fellowship and program, and was a 2010 Google Scholar. As a STEP fellow, Mr. Matthews served as a student teacher in Westlake High School in Atlanta, GA. Also, while at Georgia Tech, Mr. Matthews was an active member of the Black Graduate Students Association, a graduate mentor in the Opportunity Scholars Program, and Head TA for the Introduction to Signal Processing Course in the ECE department.

While a student at Rensselaer and Georgia Tech, Mr. Matthews took on numerous internship and extended co-op assignments with Lucent Technologies in Andover, MA; Sikorsky Aircraft in Trumbull, CT; Texas Instruments in Stafford, TX; the IBM TJ Watson Research Center in Yorktown Heights, NY and MIT Lincoln Labs in Lexington, MA. Mr. Matthews is a member of the IEEE.